CS4FN

Computer Science for Fun

The Annual Issue 1

Queen Mary
University of London

(Print) ISSN 1756-1868
(Online) ISSN 1756-1876

Welcome to cs4fn

Welcome to the cs4fn Annual: The best of the fun side of computer science.

cs4fn is now over 2 years old, so we decided to celebrate by pulling together the first three, sought after but hard to get, issues into a single Collectors Edition Annual.

We've also added some extra, classic articles from the webzine.

We are passionate about all things to do with computer science – we hope some of our passion will rub off on you. If you enjoy this issue then check out our webzine **(www.cs4fn.org)**, which has lots more stories, puzzles and links to other interesting places.

What is computer science about? We think it's about fun, more than anything...and lots of computer scientists would be doing it for a hobby if they weren't being paid for it! Come to that, an article in Popular Electronics magazine inspired Bill Gates and Paul Allen's first product: Microsoft started out as a hobby in their garage.

Computer Science's touch pervades the way everyone has fun. The most visible link is in the film industry where the computer-generated imagery used is based on leading edge research in computer graphics and artificial life. Pop concerts and TV shows like Big Brother can be streamed live round the world thanks to computer science. Listen to your MP3 player or use your mobile phone and you are using Computer Science – but Computer Science isn't about using technology, it's about creating the technology that comes next. The next step in the mobile revolution is wearable computing - clothes with computing power. What comes after that? Maybe you will play a part in creating that future.

Computer Science is about magic, but not the Hogwart's kind. The great thing about technology is that it is better than magic. Magic is about the impossible. Computer Science is about making the impossible routine. Read on to find out more.

Linus Torvalds was a student in Finland, when he posted the newsgroup message:

'Hello everybody out there ... I'm doing a (free) operating system (just a hobby, won't be big and professional ...).'

Linux, the operating system he created, is part of a $14 billion industry that Microsoft now view as their number one threat.

CS4FN
Computer Science for Fun
Summer 2005
'Clever' lumps of silicon
The Compute-Ability Competition
Computer Science at the movies
Sodarace Special
Create artificial life forms

Why computer science?

'You can do things that can't be done in reality – anything is possible.'
Oussama

'It was my hobby and I was heavily into gaming.' **Lukasz**

'Programmers work programmers' hours – whatever you want them to be.'
Tim

'It's fun writing software that you then see other people using.'
Andrew

'It's just fun and [you can use your] imagination – you can make things that never existed before.'
Nick

'…to grow intellectually and to have loads of fun at the same time.'
Naresh

'…the diversity. It doesn't close any doors.' **Hayley**

'When leaving college I was sure of one thing, I loved all things computer science.'
Adam

Enter the maze...

Most websites are designed to be easy to navigate – after all you want people to be able to find things. But what if you really want people to explore your website by wandering from page to page, finding interesting things that they weren't really looking for? What happens if you ignore the normal rules about making things easy to find?

This is what the *cs4fn* website does. Computer systems are often designed around metaphors – like the Microsoft Office desktop on your PC. It's not really the top of a desk but by designing it to look a bit like one, it makes it easier for people to use, as they can guess what they might be able to do, based on what they do on a real desktop. In a real office you move 'documents' around, open them, put them in 'files' or the 'recycling bin'; the metaphor suggests that you can do the same on your PC. For the *cs4fn* website, we wanted people to 'explore' it, so we have used the metaphor of a 'maze of rooms'. Just like a real maze, you wander from place to place, down dead-ends, passing through rooms with interesting things in, where you might stop for a while, never knowing quite where you are or what is coming up next, but having fun just trying to find the centre of the maze.

Explore the *cs4fn* website **(www.cs4fn.org)** through the maze…

'The maze is a great idea' – Adam, Computer Science undergraduate

Is that my equation ringing or yours?

'That ringtone sounds like Dr Who'

Mathematical equations make some people shudder just to look at, whereas mathematicians claim to see beauty in equations. Most people agree music can be beautiful (though you may argue between Mylo and Miles Davis, Beyoncé and Beethoven or Dr Dre versus Debussy).

Beauty, maths and music are closely related. People think maths is just about numbers but it is really about patterns. An equation is just a precise way of describing a pattern – and computer science is partly about actually using those patterns to do something useful. Music is made from patterns of sound – different kinds of patterns lead to different musical styles.

If maths and music are both about patterns, we wondered if it would be possible to listen to mathematical formulae.

Does that make you wonder what an equation sounds like? It made us think and, being computer scientists, we also wondered what we could do with it… so not only have we made it possible for you to hear what a piece of maths actually sounds like, now you can download mathematical ringtones for your mobile – for free – from the *cs4fn* website **www.cs4fn.org**

Brain the size of a planet

The War of the Worlds? Could it happen? Are there Extraterrestrials out there, watching and waiting, planning to invade Earth? Your computer could help the world find out.

When you go to bed at night your computer doesn't need to sleep, and neither do hundreds of millions of other computers all over the world. All that computing power going to waste while you snooze, seems a shame. And that's where a new technology for the 21st century comes into play.

It's called GRID computing and it's heading to be the next big thing, now that the World Wide Web is such a part of our everyday lives.

You might have already played with some of the ideas behind grid computing, if, for example, you've been running the SETI (Search for Extra Terrestrial Intelligence) screen saver **(http://setiathome.ssl.berkeley.edu/)**. The SETI group is based at Berkley, in the USA. They are trying to find signs of alien broadcasts amongst the stars. The problem is that there are a lot of stars and heaps of data from radio telescopes to work through to find those elusive signals. The solution? A screen-saver that pops up when you pop off to have a cup of tea, and uses your computer's unused computer power to search some of the radio telescope data. Like a normal screen-saver, it stops when you come back refreshed but perhaps that cuppa could go down in history when **your** computer finally finds ET.

So that's the idea behind GRID computing: all over the world, the Internet harnesses spare computer power to do big, difficult data processing jobs. GRID software allows you to turn computers in different parts of the world into 'computer clusters', where the software decides on the best way to distribute and schedule the work over the member computers, based on the clusters' power and availability. There are even plans to be able to charge companies for this service – you could let your computer join the GRID cluster and the company whose data you were searching would pay you for the 'work' done by your PC.

There are still lots of interesting issues left to solve, such as writing even faster, better GRID software, as well as dealing with security problems. Suppose a bank wants to process lots of sensitive financial data on a GRID cluster, which means that part of that data will be pumped into your PC. What's to stop less scrupulous folk taking a peek at the data?

As with all great software projects, there are tough technical problems to be overcome. But add to this further problems associated with different laws in different countries about what kind of data can be processed and transferred, combined with a need to make it all secure from criminals, and you can see that the development of GRID software will prove to be an interesting worldwide challenge. It will keep computer scientists busy for years. Perhaps in the future you will be the one who helps to make the GRID happen.

Past, present, future

Past

The first instance of 'spam' (unsolicited email) is believed to have been an announcement of a product presentation sent on 3 May 1978 by a Digital Equipment Corporation salesman to several hundred scientists and researchers on the ARPAnet – the original version of the Internet.

Present

80 per cent of text messages currently received in Japan are unsolicited junk texts ('spim').

Future

Watch out for 'spouch' or 'touch spam'. 'Haptic' interfaces – Interacting with computers by touch rather than sight – is a hot area of research right now. Mobile phones that vibrate silently in your pocket are a very simple version of a haptic interface. Sending sensations over the Internet is an obvious thing to do with it – sending people digital hugs rather than texts – and the spammers won't take long to catch on.

The future may also hold lots of 'spit' in store (spam on Internet telephones). It won't be long before Internet phones are commonplace: phones that use Wi-Fi networks to allow people to talk over the Internet rather than by using the expensive telephone networks. At that point the spammers are likely to start sending junk voicemails…

When do you think the first database was built? 5 years ago? 15 years ago? 50?

In actual fact, one possible contender for the first database is a book created by Saint Isidore of Seville. His 20-volume book *Etymologiae* aimed to be an encyclopedia of all knowledge 1,400 years ago, covering subjects like grammar, geometry, law, military history, agriculture, public games and furniture.

Etymologiae was structured in a way very similar to a modern database, hence the claim to be the creator of the first database. He drew his information from a vast number of sources, and accepted all the 'facts' collected unquestioningly. *Etymologiae* was very much like the web in that readers have to make their own judgements – he included both reliable and unreliable information for his readers to choose from, as a search engine might for you.

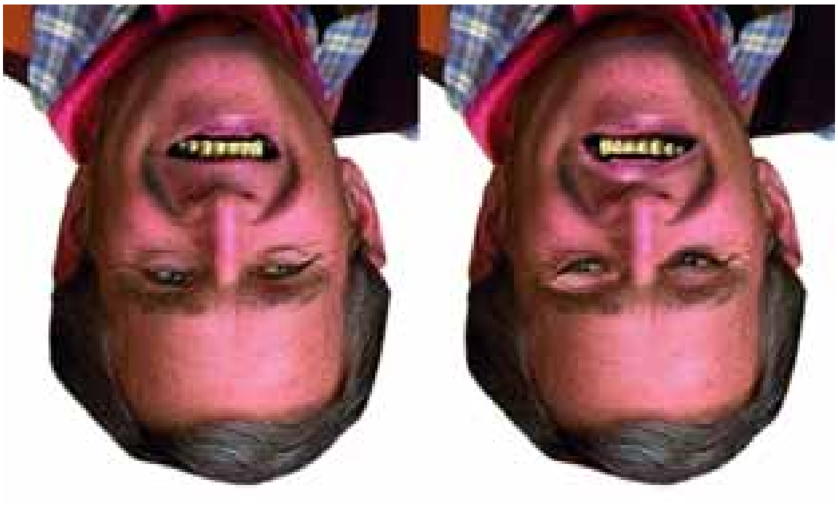
Computers follow rules – so do our brains

Like a computer, our brain uses 'rules' to help us understand the world. Often we don't know that our brain uses these rules, and it's only when the rules break down that we notice this. This is where optical illusions come in.

In an optical illusion, the human brain makes a mistake in understanding the world. In the example here, you can see two normal upside-down faces. Not such a strange situation. Your brain has rules for recognising faces, which are very useful. But because it's not used to seeing faces upside down, the rules your mind uses to 'put a face together' are not as easy to apply. So, if you turn the page upside down, you suddenly see that your brain was confused in checking the rules. You'd never have made that mistake if the faces were upright!

That means that if you want to become a computer scientist, it helps if you understand the way that people's brains work, so that you can write program rules that let the computer cope with people. One ongoing challenge is to write rules that make computers not only behave as cleverly as people when involved in solitary activities like chess but that can also behave like good team players – working to our strengths and avoiding the weaknesses that result from our brains applying the wrong rules.

Computer science is about people too!



Do you think that the pictures have just turned George Bush upside down? Turn the image around to see!

The Matrix Reloaded – sorry to bug you

In The Matrix Reloaded (2003) Neo, Morpheus, Trinity, and crew continue their battle with the machines that have enslaved the human race in the virtual reality of the Matrix.

To find the Oracle, who can explain what's going on (which, given the twisty plot in the Matrix films, is always a good idea), Trinity needs to break into a power station and switch off some power nodes so the others can enter the secret floor. The computer terminal displays that she is disabling 27 power nodes, numbers 21 to 48, but that's actually 28 nodes: a computer that can't count and shows the wrong message.

Sadly there are far too many programs with mistakes in them. These mistakes are known as bugs because back in 1945 Lieutenant Grace Hopper, one of the women pioneers of computer science, found an error caused by a moth trapped between the points at Relay 70, Panel F, of the Mark II Aiken Relay Calculator being tested at Harvard University. She removed the moth, and attached it to her test logbook, writing 'First actual case of bug being found', and so started the term 'debugging a computer program'. As the Oracle would no doubt say 'Check for moths Trinity, check for moths'.

Sodarace: Racing artificial creatures

Humans vs. machine intelligence has been the stuff of many a good Hollywood movie. Sodarace gives you the chance to play with the ideas for yourself, along with thousands of others. Will you be able to create a creature that outruns those designed by others on the net? And if you can, how about beating those created by machines using Artificial Intelligence (AI)?

Sodarace is a joint effort between Queen Mary's Department of Computer Science and the London-based digital arts company Soda Creative Ltd. It allows people worldwide to pit their wits against machine intelligence in an online 'Olympics'. Humans, and artificial intelligence computer programs, use the free BAFTA-winning online Sodaplay constructor kit (available at www.sodaplay.com) to create lifelike virtual racers out of masses and springs, then race them over 2D terrains.

Using the tutorial websites and forum pages you can join the worldwide Sodarace community and learn how to build racers yourself, or try your hand at writing artificial intelligence programs to beat your friends.

Enjoy a day at the Sodaraces and play with the limitless possibilities in creative art and science that Sodarace allows. To start creating your own racers, follow the instructions on www.sodarace.net or follow the 'Humans and AIs Compete' links from the *cs4fn* website.

It walks!

The first few Sodacreatures you make will probably collapse in a heap or wiggle and jiggle themselves exactly nowhere. Mine did, at any rate. Making my first creature, which actually managed to lurch its way from one side of the screen to the other, was a real buzz. It walks! It walks!

That is when the fun really starts – when you start to experiment and gradually discover there are actually lots of ways you can make your creatures move – just like real-life creatures, or even babies: some crawl, some bum-shuffle, some roll their way around before they learn to walk – just like your first Sodacreatures. To get you off to a flying start (well maybe a lurching one), see the step-by-step instructions on how to make one of the simplest creatures (ones that can move from one side of the screen to the other) on the *cs4fn* website: **www.cs4fn.org**

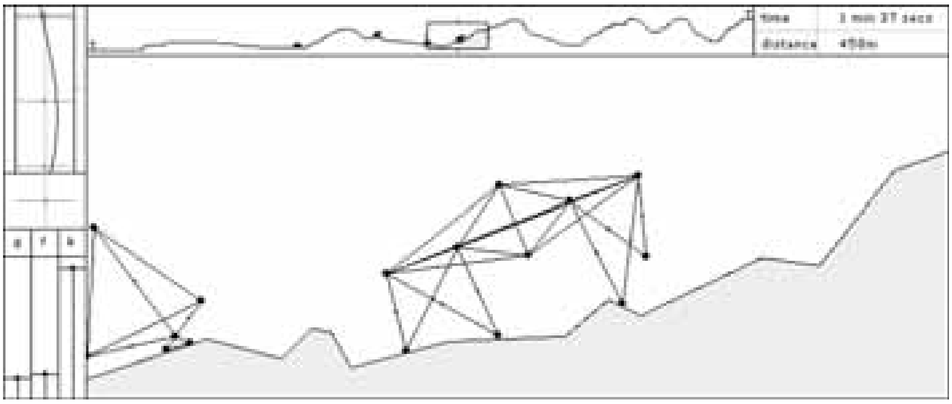
NASA TETwalkers

It's quite easy to get a triangle to lurch along using Sodaconstructor, and many other Sodacreatures are based around flexing triangles. If you think Sodaracing is just for fun though, think again. NASA is working on the concept of TETwalkers: **(http://www.space.com/imageoftheday/image_of_day_050404.html),** robots that have a lot of similarities to a 3D Sodatriangle. The trouble with walking your robot on Mars, is that if it falls over there is no one there to pick it up. TETwalkers get round that – they move around by falling over.

How should robots walk? Walking upright on two legs is not as easy as it seems to you. Watch a baby try! The animal world has come up with many other ways of doing it, and science fiction films are catching up. Early film robots tended to look like humans, though now they are just as likely to scuttle as in Minority Report or the new Doctor Who.

Moving by falling over is a fairly novel way to do it, though. TETwalkers are pyramid shaped frames. Just like in Sodarace, the edges can flex. If you flex the TETwalker edges in the right way, the centre-of-gravity of the top of the pyramid makes the whole thing fall over – the walker is now a bit further along and ready to move again, with a different point at its apex. TETwalkers have already been tested in Antarctica.

NASA is planning to create swarms of TETwalkers that are connected together and move like a snake or an amoeba – and that really does sound like Sodarace. Maybe you can out-invent NASA and build a Sodacreature that could be the basis of a Mars Explorer?



'It's life Jim, but not as we know it!'

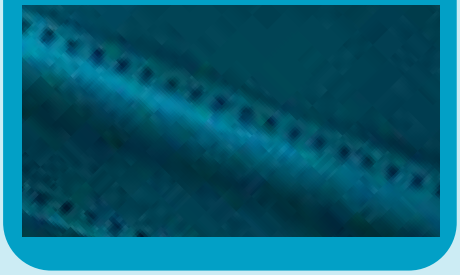
Pop down to your chemist and buy some oxygen, carbon, hydrogen, nitrogen, calcium, and phosphorus. OK, now put it all together to make a human being. What's the problem? All the necessary chemicals you need are there, it's just a question of knowing how you put them together.

Life as we know it comes from the right combinations of chemicals, but what do we mean by life? It may seem odd that computer scientists ask this question; surely it's a question for biologists or philosophers? Well, computer scientists can be both, and in the fascinating field of artificial life (A-life) they explore these very questions.

Artificial life asks questions about how 'life as we know it' and – possibly more interestingly – 'life as it could be' come about. Through writing computer programs or building robots to simulate (to try and duplicate or model) real living biological systems we can start to try and understand how our biology works. We can also model the process of evolution in computers, so that we can build our own alien worlds and populate them with our own creatures, and see how they change and adapt to survive. We could, for example, throw in asteroid strikes to change the climate and see what happens next.

Film facts

The first film shot entirely on digital cameras was 'Star Wars: Attack Of The Clones', directed by George Lucas in 2001.



Some A-life computer scientists work in the field of xenobiology, trying to work out how aliens on distant planets might look, or use some of the A-life techniques to build better robots to help us work and play. We can even put these life-like qualities into software 'agents', small programs that wander around the web collecting data – that's where the idea for Agent Smith came from in The Matrix.

A-life helps us to get a better understanding of living processes. Some scientists, who believe in so-called 'strong A-life', even argue that they are not just simulating life in a computer, but actually creating it – and that life doesn't just belong to us carbon-based life forms. Lots of deep philosophical questions to argue about there!

We also use A-life in movie special effects, computer games and TV. It's used when thousands of Orcs are animated, or to show passengers wandering on the deck of the Titanic. It is used to clone warriors in battle, and, in Doctor Who, to show mechanical spider creatures crawling over Platform 1, chasing the Doctor. The computer animator doesn't tell each creature what to do; they interact and behave by themselves using A-life techniques, giving us very life-like performances (in some cases better than the human actors!).

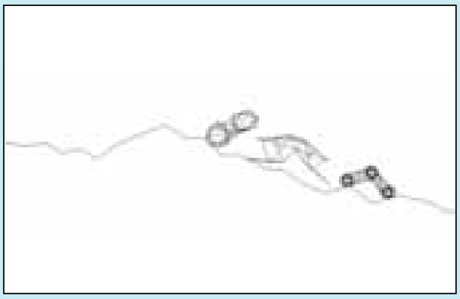
With all of this going for it, A-life (not surprisingly) is always a popular topic for student projects. The picture shown is of one creature evolved on a simulated world, where a whole range of different creature shapes occurred, depending on climate and the environment. The student who created this creature didn't claim that it was 'alive', but this is a good example of 'life as it could be' in action.

So the next time you watch The Matrix or see hordes of computer-generated monsters rampaging on the screen, spare a thought for the computer scientists behind the scenes. A-life is their life.

Sodarace – Your starter kit for ten

1 Look at the Sodarace website

Go to www.sodarace.net and www.sodaplay.com and have a look around. In particular, look at the 'How do I start?' sections on www.cs4fn.org. Sodarace provides you with a set of powerful tools to explore your creativity, and also to compete with AI-designed racers. There are also loads of ways to use Sodarace in school: there are even lesson plans in the forums, so get your teachers interested and you could be Sodaracing in lessons!



2 Explore the Sodaconstructor website:

To build Sodarace creatures you use Sodaconstructor, so you need to learn how to use this online tool set. It's really quite simple, but it will take time to become a skilled Sodaconstructor. The cs4fn site has step-by-step instructions with screen dumps to get you started on building a very simple creature that walks. Have a look at the Sodaplay constructor: www.sodaplay.com/constructor/

3 The building blocks

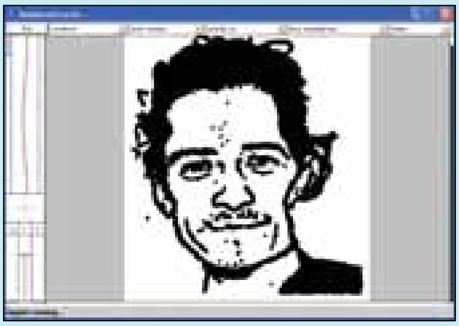
All the structures built in Sodaconstructor are made of three basic components: masses, springs and muscles. In Sodarace:

- **Masses** are single points that have weight and obey the laws of gravity.
- **Springs** connect the masses.
- **Muscles** do the moving.

You have two 'modes':

- the **'construct'** mode where you can draw your models, and
- the **'run'** mode where the muscles do their stuff and start to move.

You go back and forward between these modes using the buttons at the top of the screen.



4 Change the world

The Sodaconstructor program allows you to build a creature by pointing and clicking your mouse on the screen (right-hand mouse click to stop). At the right-hand side of the screen is the control panel, and at the bottom there are three slide controllers – these let you change the laws of the world your creature lives and moves in. You can change gravity, friction and the stiffness of the springs.

- **Gravity** (g): If you turn the gravity up, your models will be squashed by their own weight. Turn it down low and your model will float. You can even turn gravity upside down using the popup menu.
- **Friction** (f): Friction slows moving masses. Apply lots of friction to your model and it will look like it's moving in treacle. Reduce friction and your creature can move fast but it might wobble out of control.
- **Springiness** (k): This sets the springiness of your model's springs: weak springs make your model go floppy, very stiff springs are strong, but can make your model too jittery.

5 Making muscles move

You can turn any spring into a muscle by attaching it to the energy wave that repeatedly gives it pulses of energy. Remember that only muscles can power your creations to make them move.

Switch from construct to select. When you click on the spring you want to 'muscle-ise' you will notice that a small circle appears around the centre point of the spring, and also that on the bottom of the screen there will be a short line below the wave that also now has a circle. Using your mouse, move this line (it will control the muscle you've selected) to some point on the wave.

When you switch to 'run' mode the wave will start to move, and as the wave passes through the muscle control line you have added, the muscle will expand and contract in time with the wave passing through.

You can build graceful moving creations by having muscles work in a synchronised way. You can make the wave move faster and change its strength (ie the amount that the muscles expand and contract by) using the sliders on the control panel. (See 'It walks!' in the webzine for tips on how to build a really simple creature, to get you started.)

6 Saving and sending models

You've started to create, so you will want to save your work. You need to register with the site to allow this, but that's free and easy to do. The first time you use it, press the 'file' button at the top of the Sodaconstructor window) followed by the 'login'. New users register. You must be logged into your Sodaplay account to save your models. To save the model, press the 'file' button then press the 'save' button and choose a name for your creation. You can also send saved models to your friends (see the Sodaconstructor site for details).

7 Loading models

To load a model, you press the 'file' button (in the top left of the Sodaconstructor window) to switch to the browser interface. This will give you a box where you can type in the name of the model you want to load. There are plenty to choose from on the Sodaplay site! Remember that when you load a model it replaces the current model you have on screen, so save the current model first if you want to play with it again later.

8 Read the tutorial sites and ask for help

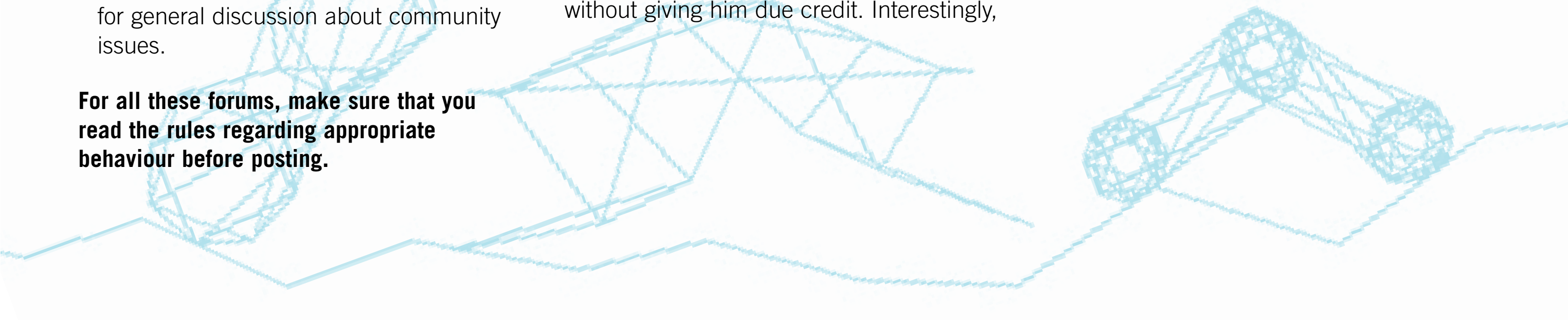
You now have the Sodaconstructor basics, the rest depends on your skill and a little help from the worldwide Sodarace community. There are forums where you can ask other constructors for help. Please use this facility sensibly. It's a brilliant resource for help, so don't post silly messages – the forums will not like it! The tutorial sites are full of expert information about basics and advanced constructing. You can also see some of the wonderful creations others have made, which should inspire you to create more Sodacreatures.

9 Know your forums

There are currently three user forums at the centre of Sodaplay, each with a different theme.

- The **Sodaplay** forum is where you talk about the website and its many applications including Sodarace.
- The **Sodaplay support** forum is the place to share your model-making tips and interact with the Sodaracing community.
- Finally the **random talk** forum is the forum for general discussion about community issues.

For all these forums, make sure that you read the rules regarding appropriate behaviour before posting.



10 Ready to race

Once you have a racer (a moving soda creation), you might want to learn to race it. The race software requires you to download the correct Java plug-in for your browser. The standard Microsoft version of Java doesn't have all the abilities needed, so follow the instructions on the Sodarace home page to install the correct version of Java.

Once the new Java is installed, check by going to Sodaplay/create and opening a race.

There are lots of races previous users have built. The race application lets you load racers and terrains and modify them. Amoebamatic uses Artificial Intelligence to create wheely racers which you can customise.

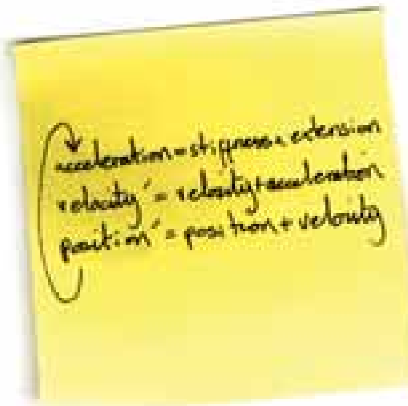
Getting to grips with the race application tool takes a little practice, but the forums are there to get you going. It will be worth the effort. Why not share your experience with others in the forum or set up your own tutorial website? That's the Sodarace philosophy: 'Tools not Rules'. It's up to you to explore and push the races to the limit!

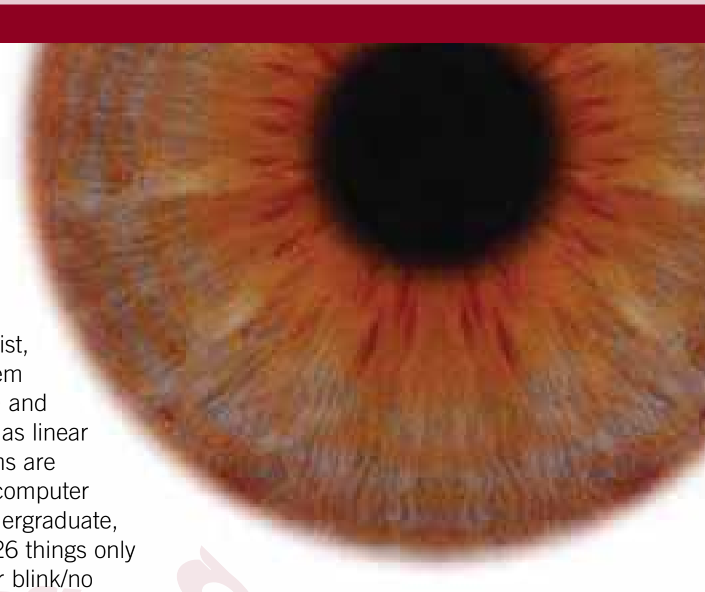
Hooke versus Newton

The laws of Physics that form the basis of Sodarace are so simple that they fit on a Post-It™ note. That is the beauty of Physics. The springs in Sodaconstructor work using rules discovered by British scientist Robert Hooke (1635-1703), who found that the force a spring exerts is directly proportional to how much it extends. This is called Hooke's law (Discovering a law of nature is a great way to be immortalised). Hooke also invented the iris diaphragm in cameras, the universal joint used in motor vehicles, the balance wheel in a watch, was the originator of the word 'cell' in biology, helped develop the microscope and worked with Sir Christopher Wren in rebuilding London after the Great Fire of 1666. Not bad going really. He also had a famous feud with Sir Isaac Newton (1642-1727), as Hooke felt that Newton had taken his ideas without giving him due credit. Interestingly,

Sodaconstructor uses the laws of motion discovered by Sir Isaac, so Hooke and Newton work together now in Sodaconstructor at least.

Newton's work on light and gravity are well known, but he was also an alchemist, a member of parliament and Warden of the Royal Mint. He successfully foiled the coin counterfeiters, who were running riot at the time, by exchanging all the coins in the country for better designed ones: a task that required his enormous attention to detail and ability to improve the way people did tasks: both skills that would have made him a great programmer. He also set up a spy network worthy of Walsingham (see page 12) saying of his foes 'Criminals, like dogs, always return to their vomit'. Newton also invented the cat flap… the result of kittens ruining his optics experiments.


acceleration = stiffness x extension
velocity' = velocity + acceleration
position' = position + velocity



Locked-In syndrome

One of the worst medical conditions must surely be Locked-In syndrome. It leaves you with all your mental abilities intact but totally paralysed, except perhaps for the blink of an eye. A perfect, working mind is locked inside a useless body: the sufferer can sense everything around but is unable to communicate with anyone. Despite this, one of the most uplifting books I have read is *The Diving Bell and the Butterfly*. It is the autobiography of Jean-Dominique Bauby, written after he woke up in a hospital bed with Locked-In syndrome. In the book, he describes a life with Locked-In syndrome, including how he communicated, not only with medical staff, friends and family, but also how he came to write the book without any technological help.

The book was written using a heroic form of face-to-face interaction. Put yourself in his position, waking up in a hospital bed. What would be the best way for you to write a whole book? You have only a helper with a pen and paper to write down your 'words'. The only movement you can make is to blink your left eyelid.

How did Bauby do it?

Bauby's helper read the alphabet aloud ('A, B, C…') When the letter he was thinking of was spoken, Bauby blinked. The helper would write that letter down and then start again, letter after letter. Try it with a friend – communicate your initials to them that way. Now imagine that that is the only way you can talk to anyone. I hope your name isn't Zebedee Zacharius Zog or Zara Zootle.

Bauby realised that the 'A,B,C' method could be improved upon. He had been the Editor-in-chief of the French women's magazine *Elle* before he became ill, so he knew about language. He knew that some letters are more common than others in natural language, so he got the helper to read out the letters in order of frequency in French, 'E...S...A...R….' That way the helper got to the common letters more quickly. A similar trick has been used through the ages to crack secret codes (see the Beheading story on page 12) and for doing the crossword-like puzzles called cross-references (try one on page 18).

Now, as a computer scientist I immediately start to think that I could have made his life so much better (even without replacing the human helper with blink detection gadgets and the like). In the worst case, perhaps dictating a story where someone snores 'Zzzzz', would take 26 questions per letter.

On average, in the course of dictating the whole book, roughly 13 letters will be said per letter dictated. Bauby's modification improves things but the worst case is still 26. Thinking as a computer scientist, the problem is a search problem (searching for one letter in 26) and the solution he used is known as linear search. Other search algorithms are far better. From some simple computer science that I learnt as an undergraduate, I know that a search through 26 things only needs at most five true/false or blink/no blink questions – not 26.

Learning from a children's game

How do we do it? By using the same strategy as is used in the children's game of 20 questions. It is a search problem too – a search to find the name of a famous person out of thousands – and yet it does not take thousands of questions to win. Played well, you do not ask as the first question 'Is it Nelson Mandela?', the equivalent to 'Is it E?' Rather you first ask: 'Are they female?' and so rule out half the possibilities whatever the answer. The equivalent question for the alphabet is 'Is it before N?' Try it – start with 1 million and see how many times you have to halve it before you get down to one. 1,000,000 … 500,000 … 250 000…

A similar trick has been used through the ages to crack secret codes

Keep asking questions like that about letters rather than famous people and you get down to a single letter in no more than five questions. Tweak it based on letter frequencies and you can do even better for the common letters.

Bauby should have got the helper to ask such halving questions. Think about it. Five questions at worst rather than 26, multiplied up by all the letters in his book. If only he had known some computer science, how much easier his life would have been.

Now we have worked out a method we can think how we could automate it with suitable technology. How wonderfully computer science can improve lives.

But wait a minute. Perhaps the computer scientist would have ensured his book was never completed and his life was even more a hell. Perhaps we should have started with the person rather than our bright ideas. What if blinking is a great effort for him? His solution involved him blinking only once per letter. Ours requires him to blink five times. Multiply *that* by a whole book. Furthermore, his solution is easy for anyone to walk in and understand. Ours is complex and might need some explaining before the visitor understands and Bauby is not going to be the one to do the explaining.

It worked for him!

One thing is certain about Bauby's solution – it worked for him. He wrote a whole book that way, after all. Perhaps the helper did more than just write down his words. Perhaps they opened the curtains, talked to him about the outside world or just provided some daily human warmth. Perhaps the whole point of writing the book was that it gave him an excuse to have a person there to 'talk' to all the time.

Replace the person and perhaps you have replaced the one thing that was actually keeping him alive. In an extreme 'usability situation' such as this, the important thing is that the user really is involved throughout the process. They are the ones who ultimately have to make it work for them, not only technically but also emotionally and socially. Otherwise we may devise a 'solution' that is in theory wonderful but in practice hell on earth for the user.

As you can see, computer scientists have to think about so much more than just computers.

Find out about Jean-Dominique Bauby and what life is like with Locked-In syndrome by reading: *The Diving Bell and the Butterfly* by J-D Bauby, Fourth Estate.

How do computers become so clever?

Computers do some miraculous things. A computer can beat the world champion at chess, even fly a plane more skillfully than a human. How do they do it? How can a lump of silicon and wire appear to be cleverer than a human? Everything that a computer does that is intelligent is ultimately down to a person – a computer scientist in fact – writing clever instructions: rules to be followed. Everything you have ever seen a computer do, was just the result of it obeying the rules written by a computer programmer years earlier. Even a piece of paper can play these games as well as humans if it contains such rules. In fact:

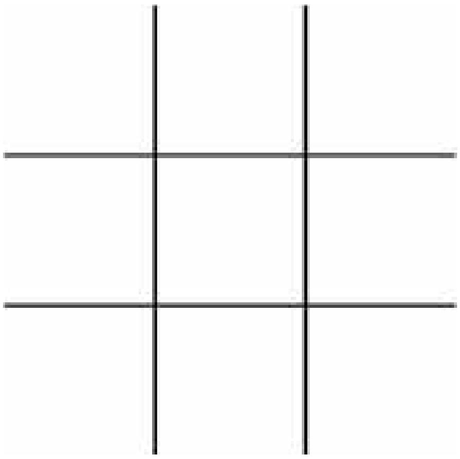
This piece of paper (let's call it Pete) can play better noughts and crosses than you can!

You will struggle to beat Pete at any rate! Try.

The following instructions give you a list of the moves that Pete the paper makes. If you are playing against Pete, you make whatever move you want to make after each one of Pete's moves. Pete gets to go first and is X. Follow his instructions as they are written.

Quick facts

By the beginning of the 1990s, a Hallmark greeting card embedded with a microchip that allowed the card to play 'Happy Birthday' contained more computing power than existed on the entire planet in the early 1950s. The one I got for my last birthday played 'Agadoo'…Arrghhhh!



Pete's Move 1:
Draw an X in a corner for me.

Your Turn:
Go where you like.

Pete's Move 2:
If no-one went there already then draw an X in the opposite corner to my move 1.

Otherwise put an X in a free corner for me.

Your Turn:
Go where you like.

Pete's Move 3:
If there are two Xs and a space in a line (in any order) then put an X in that space. I win!

Otherwise if there are two Os and a space in a line then put an X in that space. Ha!

Otherwise put an X in a free corner for me.

Your Turn: Go where you like.

Pete's Move 4: If there are two Xs and a space in a line (in any order) then put an X in that space. Gotcha! I win!

Otherwise if there are two Os and a space in a line then put an X in that space for me.

Otherwise put an X in a free corner.

Your Turn:
Go where you like.

Pete's Move 5:
Put an X in the free space for me.

Why not use Pete's moves above when you next have a game with your friends.

Obviously, if you're playing against friends, Pete's moves are the moves you make and your friend will make the moves labelled 'Your turn'. You will be invincible.

This is all a computer program is – a list of instructions that the computer can follow. The instructions for the computer have to be written very precisely in special languages so that the computer can follow them without understanding them, but the idea is the same.

Computers can only do things that the programmer has thought of – if things aren't as expected it won't seem so clever. I wrote the above rules expecting the paper to go first but what if it has to play second? Does it still seem so clever? That is the skill of the programmer: writing rules for every eventuality. Have a go at writing some better instructions for player 2 at noughts and crosses.

You can also program your own Noughts and Crosses Artificial Intelligence– see the webzine www.cs4fn.org

A puzzle, spies ... and a beheading

A puzzle about secrets

Sisters Amy and Elinor live together. Their cousin Emma wants to send them secret messages. She doesn't want Amy to read her messages to Elinor and vice versa. Amy buys them all small lockable notebooks for Christmas. They are normal notebooks except that they have a lock that can be locked shut using a small in-built padlock. The padlock can be opened with a single key. Amy suggests that they write messages in their notebook and post it and the key separately to the person that they want to send the message to. After reading the message that person tears the page out and destroys it, then returns the notebook and key. They try this and it appears to work, apparently preventing the others from reading the message. They exchange lots of secrets…until one day Amy gets a letter from Emma that includes a note added on the end from Elinor: 'I can read your messages. I know all your secrets'. She has been reading Emma's messages to Amy all along. She now wants them to know how clever she has been.

How did she do it and what does it have to do with the beheading of Mary, Queen of Scots?

Breaking the system

Elinor has of course been getting to the post first, steaming open the envelopes, getting the key and notebook, reading the message (and for the last one adding her own note). She then seals them back in the envelopes and leaves them for Amy. A similar thing happened to betray Mary, Queen of Scots to her cousin Queen Elizabeth I. To find out how, read on.

A better way?

Emma suggests a solution to the problem of intercepted messages using the notebooks and keys in a similar way, but in which no keys are posted anywhere. To prove her method works, she sends a secret message to Amy, which Elinor fails to read. How does she do it? See if you can work it out before reading on…and what is the link to computer science?

Mary, Queen of Scots

The girls face a similar problem to that faced by Mary, Queen of Scots and countless spies and businesses with secrets to exchange before and since – how to stop people intercepting and reading your messages.

There are two ways to make messages secret – hide them so that no one realises that there is a message to read or disguise the message so that only people in the know can read it, or both. Hiding the message is called *Steganography.*

Disguising a message so that it cannot be read is called *encryption.* The girls in the puzzle discovered, just like Mary, that weak encryption is worse than no encryption, as it creates a false sense of confidence that the messages are secret.

Mary, Queen of Scots ultimately lost her life because her encryption was easy to crack. She believed the encryption would protect her. It had given her the confidence to write what she otherwise would not have written.

House arrest

Mary had been locked up – under house arrest – for 19 years by Queen Elizabeth I, despite being captured only because she came to England seeking refuge from her cousin Elizabeth after losing her Scottish crown. Elizabeth was worried that Mary and her allies would try to overthrow her and crown Mary Queen of England if given the chance. Elizabeth thought that it would be better to lock her up before she even thought of treason. Towards the end of her imprisonment, in 1586, some of Mary's supporters plotted to free her and assassinate Elizabeth. Unfortunately they had no way of contacting Mary, as letters were not allowed either in or out by her jailors.

Mary, Queen of Scots lost her life because her encryption was easy to crack

Then the plotters had a stroke of good luck. A young priest called Gilbert Gifford turned up claiming he had worked out a way to smuggle messages to and from Mary. He wrapped the messages in a leather package and hid them in the hollow bungs of barrels of beer. The brewer delivered the beer to Chartley Hall where Mary was held and the packages were retrieved by one of Mary's servants. This (a form of steganography) was really successful, allowing Mary to exchange a long series of letters with her supporters. Eventually the plotters decided they needed to get Mary's agreement to the full plot. The leader of the coup, Anthony Babington, wrote a letter to Mary, outlining all the details of the plot. To be absolutely safe, he also encrypted the message using a cipher that Mary could read (or 'decipher'). He soon received a reply in Mary's handwriting, also encrypted that showed that Mary agreed to the plot but also asked for the names of all the others involved. Babington responded with all the names. Unfortunately, unknown to Babington and Mary the spies of Elizabeth were reading everything they wrote – and the request for names did not even come from Mary.

Spies

Unfortunately for Mary and Babington, all their messages were being read by Sir Francis Walsingham, the ruthless Principal Secretary to Elizabeth and one of the most successful spymasters ever. Gifford was his double agent – the method of exchanging messages had been Walsingham's idea all along. Each time he had a message to deliver, Gifford took it to Walsingham first, whose team of spies carefully opened the seal, copied the contents, redid the seal and sent it on its way. The encrypted messages were a little more of a problem, but Walsingham's code breaker managed to break the cipher. The approach, called *frequency analysis*, which works for simple ciphers, involves using the frequency of letters in a message to guess which is which. For example the most common letter in English is E so the most common letter in an encrypted message is likely to be E. It is actually the way people nowadays solve crossword like code-puzzles known as cross-references that can be found in puzzle books (try the one on page 18). The trick can also be used to help people who are paralysed (see the story about Locked-In syndrome on page 10). Walsingham now had the key that allowed him to read even encrypted messages.

Can you break the code on page 18?

A beheading

When Walsingham read Babington's letter, he knew that he had the evidence to hang him, but he let the letters continue so that when Mary replied, Walsingham and Elizabeth finally had the excuse to try her too. Up to that point (for the 19 years of her house arrest) Elizabeth had not had strong enough evidence to convict Mary – just worries that Mary would be a magnet for plotters. Walsingham wanted more evidence though, so he forged the note asking for the names of other plotters and added it to the end of one of Mary's letters, encrypted in the same code. Babington fell for it, and all the plotters were arrested. Mary was tried and convicted. She was beheaded on 8 February 1587.

Private keys... public keys

Let's go back to our secret-swapping cousins. How does Emma's method get round the problem of her messages being intercepted and read? Her main weakness is that she has to send Amy the key, as well as the locked message – if the key is intercepted then the lock is worthless. An alternative way means that she doesn't have to keep sending the key to Amy. Suppose Emma wants to send a message to Amy. She first asks Amy to post her notebook (without the key but left open). Emma writes the message in Amy's book then snaps it locked shut and posts it back. Amy, who has kept the key safe all along, opens it secure in the knowledge that the key has never left her possession. This is essentially the same as a method known by computer scientists as *public key encryption* – the method used on the Internet to protect the exchange of messages and that allows the Internet to be secure. In this scheme, keys come in two halves: a 'private key' and a 'public key'. Each person has a secret 'private key' of their own that they use to read all messages sent to them. They also have a 'public key' that is the equivalent to Amy's open padlock. If someone wants to send me a message, they first get my public key, which anyone who asks can have. It is used to encrypt the message (close the padlock), but is no use to decrypt it (reopen the padlock). Only the person with the private key (the key to the padlock) can get at the message. So messages can be exchanged without the important decryption key going anywhere. The message can't be intercepted.

Would this have helped Mary, Queen of Scots? No. Her problem was not that she exchanged keys but that she used a method of encryption that was easy to crack – in effect the lock itself was not very strong and could easily be picked. Walsingham's code-breakers were better at decryption than Babington was at encryption.

If you want to find out more about spies, encryption and even some of the computer science behind it then why not read *The Code Book* by Simon Singh, Fourth Estate.

Computers that work to our strengths?

One important thing about computer programs is the way that they present information to the people using them. In the early days of computing, people interacted with computers by typing commands and getting written results. Nowadays computers use 'GUIs' (Graphical User Interfaces) to represent information with graphics. Used well, GUIs make a big difference to the ease of a particular job. To see how much difference the way that information is organised and presented can make, let's play a game called Spit-Not-So. Write down the words:

SPIT NOT SO AS IF
IN PAN FAT FOP

1 The first player chooses a word that is on the list and crosses it out.

2 The first player writes the word down in front of them.

3 The second player then does the same thing choosing a different word.

4 The players take turns to do this until one person wins.

The winner is the first player to hold three words containing the same letter.

An example game might go:

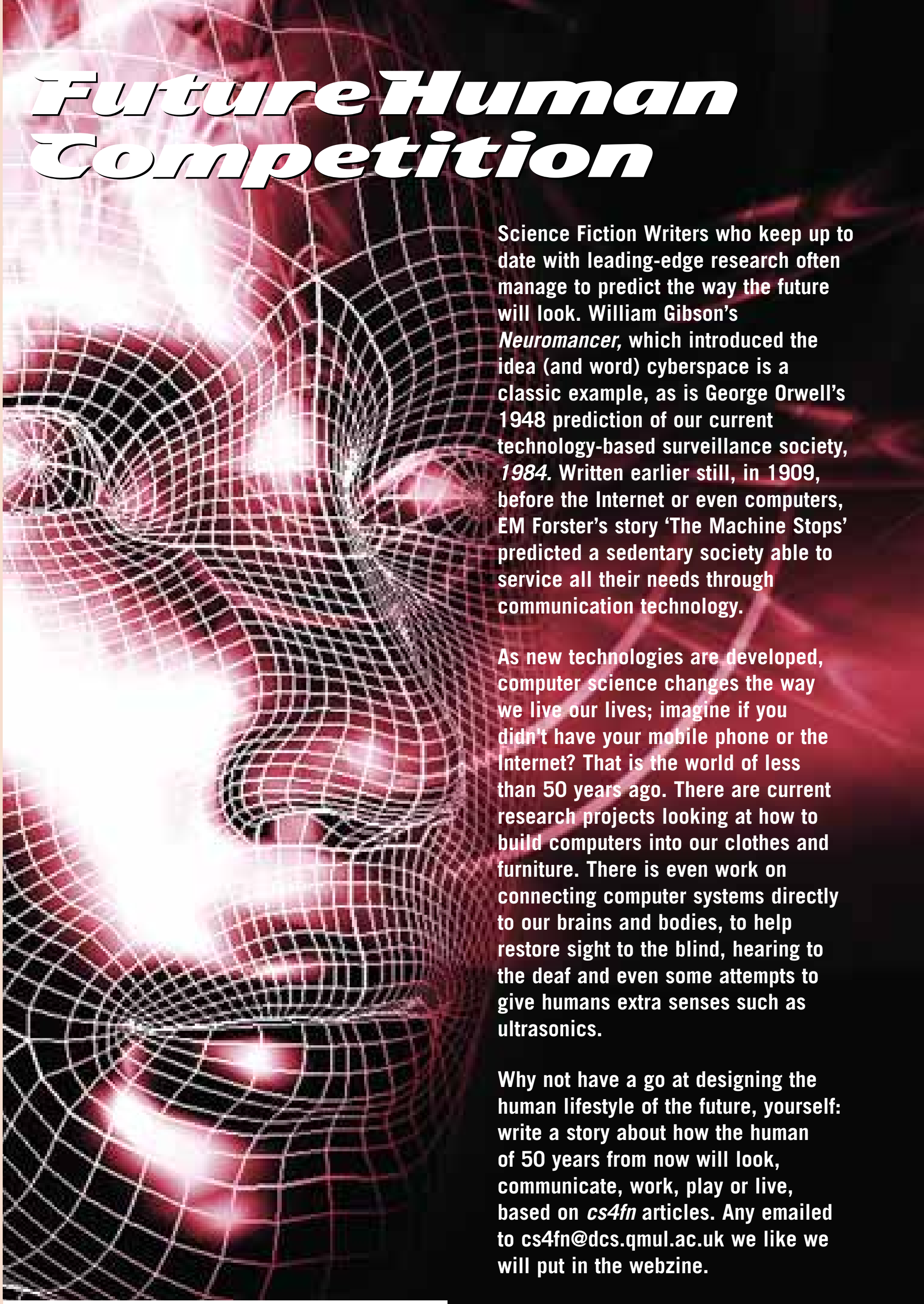
Player 1 takes NOT	Player 2 takes SPIT
Player 1 takes FAT	Player 2 takes PAN
Player 1 takes FOP	Player 2 takes IF
Player 1 takes SO	…and wins holding

NOT, FOP and SO – 3 words with 'O'.

Play a few games to get the idea, then go to the cs4fn webzine for some sneaky tips and to see the link to GUIs!

Numbers game 2

The number of letters transmitted over the Internet before it crashed for the first time. The Internet was born on 20 October 1969 with the first transmission of data. The letters L and O were transmitted but the system crashed when the G of LOGIN was entered from a computer at the University of California and sent to another one at a research centre at Stanford, near San Francisco.



Future Human Competition

Science Fiction Writers who keep up to date with leading-edge research often manage to predict the way the future will look. William Gibson's *Neuromancer,* which introduced the idea (and word) cyberspace is a classic example, as is George Orwell's 1948 prediction of our current technology-based surveillance society, *1984.* Written earlier still, in 1909, before the Internet or even computers, EM Forster's story 'The Machine Stops' predicted a sedentary society able to service all their needs through communication technology.

As new technologies are developed, computer science changes the way we live our lives; imagine if you didn't have your mobile phone or the Internet? That is the world of less than 50 years ago. There are current research projects looking at how to build computers into our clothes and furniture. There is even work on connecting computer systems directly to our brains and bodies, to help restore sight to the blind, hearing to the deaf and even some attempts to give humans extra senses such as ultrasonics.

Why not have a go at designing the human lifestyle of the future, yourself: write a story about how the human of 50 years from now will look, communicate, work, play or live, based on *cs4fn* articles. Any emailed to cs4fn@dcs.qmul.ac.uk we like we will put in the webzine.

Sodarace (see page 6) was featured on the official movie website for Terminator 3 under the headline:

'Before robots can rule the world, they have to learn to walk'

What do you think is most likely to disappear next?

Have your vote on the cs4fn site...

□ Fixed phones?

□ Cables?

□ Written signatures?

□ Loose change?

□ Wrist-watches?

□ Paper?

□ Physical shops?

□ Calculators?

□ Radios?

□ MP3 Players?

Computer science at the movies

Computer scientists develop the advanced digital technologies that make many film special effects possible. Their work and visions help shape the futuristic worlds that form the backdrop to many science fiction and fantasy films. Will criminals, for example, one day need to rip out their eyeballs to escape detection as in the film Minority Report or is biometric (biology)-based identification just a fantasy?

Stealing digits

In the future shown in the film Minority Report, biometrics – security systems based on biological features like eye scans, face recognition and fingerprints – are an everyday part of society. Other films regularly have biometric locks protecting vaults and top-secret control centres. Biometric security is not fiction though. It is already here, and not just in high-tech James Bond situations. The US is already using it at immigration desks and several makes of car have biometric ignitions. Biometrics is a popular technology, as it is so hard to forge. PINs can be stolen or forgotten and cards or keys can also be lost, but your fingers and eyes go everywhere with you – or do they?

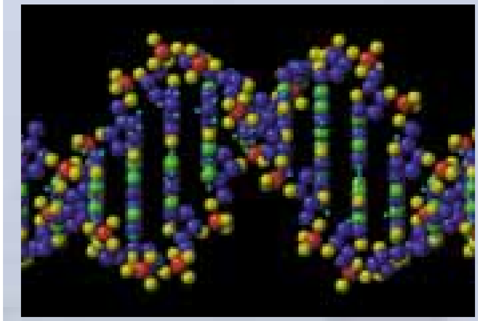
While having eye transplants, as in Minority Report, to open a lock is still a long way in the future (let alone the even more extreme situation seen in Face-Off, where Nicholas Cage and John Travolta steal each others identities by having their faces transplanted), a grisly biometric theft has already happened. Carjackers in Malaysia, on finding the car they were stealing had a fingerprint recognition system to start the engine, cut off the tip of the owner's finger and stole that too. Which makes you think twice about using biometrics that don't need a living body attached!

Computing with the code of life

Computers and DNA: 'That's an odd combination', you might say, but in the near future it may be that the fastest computers on the planet are built, not from electronic bits, but from biological ones.

Electronic computers will soon face problems. We can pack more and more computing power onto a silicon chip, but this means having to build smaller and smaller parts in the silicon, and that's technologically very tricky as there are real physical limits on how small we can ultimately go. This is bad enough when physics causes problems, but worse still, geometry is against us too. As we increase the area we use on the chip, the space along the edges of the chip grows much more slowly. This means there is no room on the perimeter to put the wires we need to get all the data in and out. So what's the solution?

One possible approach is to start to build computers from other materials. Step forward DNA. DeoxyriboNucleic Acid is the chemical stuff that our genes are made off, it's the 'code of life' that tells our cells how to work, our eyes to be brown or our hair to be curly.



DNA works because it is a long string of chemical instructions; these instructions are in an alphabet with just four letters A, T, C and G, which correspond to the names of the four chemical 'bases' that make up DNA.

In 1994, computer scientist Leonard Adleman had the brilliant idea of using DNA to solve mathematical problems. He chose to begin with the famous 'Travelling salesman problem'. In this problem, a salesman has to visit a number of towns (let's refer to this number as N) and use the shortest route to visit them all (he needed to keep his travel expenses as low as possible). This sounds simple, and it is with just a few towns, but try it when N is 100 or 1,000. This is one of those classic problems that suffers when you scale it up: there is no known way to calculate the best solution quickly for a large N value.

Adleman coded up this problem on bits of DNA for just seven cities. Different strands of DNA were chosen to represent each city, in a particularly clever way, so that when Adleman actually mixed the solutions together in a test tube, the way the DNA chemical bonds joined up solved the problem, with the chains of resulting DNA representing routes. In fact, the different, joined strands in the tube gave all the possible solutions to the problem. The only difficulty Adleman faced was going through the goo to find the answer. But he got it in the end.

Three years after Adleman's experiment, the University of Rochester developed logic gates made of DNA. Logic gates are the fundamental electronic parts normally used to build a computer. They are the parts that allow the calculations to take place. They act like tiny switches with rules that say what to output when certain signals are input and have names like AND gates OR gates or the exotic sounding XOR (exclusive OR) gates. The researchers found that they could build DNA structures that followed the rules of logic. For these logic gates, the inputs were bits of DNA rather than electronic signals, and the gate then chemically spliced these fragments together to get the single required output – a very clever bit of biology. The researchers believe that these logic gates might be combined with larger DNA microchips to create a breakthrough in DNA computing.

DNA computations are fast and accurate, and the materials used are biodegradable and cheap. There is DNA in every cell. DNA also has the ability to contain a massive amount of information. If you take one-pound weight of DNA, it could store more information than all the electronic computers ever built. It's been suggested that the computing power of a teardrop-sized DNA computer, using the DNA logic gates, will be more powerful than the world's most powerful supercomputer and, unlike conventional computers, DNA computers will be able to perform all their calculations at the same time. We call this 'parallel processing', a very interesting way to build computers.

> **'In the future, computers could be build from biological, rather than electronic, material'**

It will be fascinating to see how this new technology develops and is applied. Exciting times lie ahead for computer scientists and biologists, as they work out new ways to build us even faster computers.

Who wants to be the weakest millionaire?

TV game shows like Who Wants To Be A Millionaire? and The Weakest Link are very popular. Part of their popularity lies in the fact that they have interesting rules for the contestants to play against. These rules give the shows their tension, but someone has to make the rules up to begin with. So how would you go about designing the format for a good game show?

Cash in a box

Well, we need some prize money – the amount of cash that the player will walk off with in their pocket. Let's add a prop, a shiny box that our host puts the money into, and let's label this box with big letters 'PRIZE'. So, a very simple game show would ask the contestant a question. If they get it right, they win £100, for example, which our host puts into the 'PRIZE' box. At the end, the player gets what is in the box, so 'PRIZE' can refer to the box but also, more importantly, the value of money in the box.

Simply answer the question

How could we write the instructions for this simple quiz? Well how about:

```
If (Answer is correct)
   {PRIZE = £100}
```

We have used PRIZE = £100 to mean put £100 in the prize box. So, as we wanted, if the answer's right, the value in PRIZE goes up to £100.

Double your money!

Simple enough, now suppose that we make the game a little more challenging, so what will happen is that each time the player gets the answer correct they double their money. If they get the answer wrong then they lose the lot (a bit like Who Wants To Be A Millionaire?). So, how would we write that?

Your starter for 100?

We have a problem to begin with. If there is no money in PRIZE to start with, then doubling it when you get the answer right will give you twice as much nothing. Hardly fun to watch. So we have to put some cash in the box to begin, (or we give them a simple question to start with). Let's be kind. Let's put £100 in the box to start.

```
PRIZE = £100
```

If they get the question right we can then say that PRIZE = 2 * PRIZE. What this means is that the new money in the box will be twice the money that was in the box beforehand (we've used * for multiplication).

Twice in the box

We have our starting situation (we need something to double remember)

```
PRIZE = £100
```

and what to do if the player gets the question right:

```
If (Answer is correct)
   {PRIZE = 2*PRIZE}
```

But we were being nasty. What happens when they get the question wrong? They need to lose all the cash! That is we want to set PRIZE=£0: no cash in the box. How could we write this?

We could simply write

```
If (Answer is wrong)
   {PRIZE = £0}
```

After all the answer is either right or else it's wrong, and here's an idea about how to write this:

```
If (Answer is correct)
   {PRIZE = 2*PRIZE}
else
   {PRIZE = £0}
```

We have a single line. 'If the answer is correct, the cash doubles or else (if the answer is wrong, which is the only other option) the cash is lost.'

So for our Who Wants To Be A Millionaire?-type quiz, we have the rules.

```
PRIZE = £100

If (Answer is correct)
   {PRIZE = 2*PRIZE}
else
   {PRIZE = £0}
```

Round and round again

But we want more than one round of the game. Each additional round should become more exciting, as the prize money grows with each correct answer. How do we write this? Well we need some way to say that we do the same thing time and again, as long as we are happy to do it, but that at some stage we want to stop. Hmm. How many rounds do we want to have?

Quick quiz question

Why did a 13-year old girl from Brittan Elementary School in the USA make headlines across the world for saying: 'Look at this. I'm a grocery item. I'm a piece of meat. I'm an orange.'

Answer

She was outraged that her school had introduced electronic tags to keep a constant track of student movements within the school. As a result, the school suspended its use of the system, which just goes to show there is more to computer science than technical brilliance. To be a successful innovator, you have to understand people too.

Count up

Let's do our quiz round eight times:

```
Do 8 times {a round}
```

So we have a way to have eight rounds, and we know how to do each of the rounds, and how to start the prize fund in the box, so, let's put them together:

```
PRIZE = £100

Do 8 times
{
        If (Answer is correct)
           {PRIZE = 2*PRIZE}
        else
           {PRIZE = £0}
}
```

Does this make sense? We start with setting the contents in the prize box to £100 before we do any rounds, so that first 'PRIZE = £100' will go outside the 'Do it eight times' bit. For each round if the answer is correct the prize money doubles, and as it's the same PRIZE box we use in each round, the money will continue to double for our lucky player.

That's the wrong answer!

What happens if the contestant gets the answer wrong? Well, looking at our 'else' rule, if in any round the answer is wrong, the prize in the box goes to zero. That's a problem for our player because even if they get the next question right it would just double nothing! Unkind perhaps but would watching the contestants play for, say, five rounds, answering the questions for nothing, make good television?

Therefore, if a contestant gives a wrong answer, we want to stop the quiz and take away the prize money. The game will be over. So, let's add that to our set of rules. Put in the word 'Break' to mean just that: that we jump out of the rounds and end the game.

```
PRIZE = £100

Do 8 times
{
        If (Answer is correct)
           {PRIZE = 2*PRIZE}
        else
           {PRIZE = £0, Break}
}
```

Roll the credits

Well done. So there we have the rules for a 'Millionaire'-type game, explained more-or-less in English and fairly easy to write down. But if you've followed this through, you've actually understood your first computer program. Computer programs look just like this, a series of rules to control the way numbers (or, in our case, cash) are moved around. We call this set of rules an algorithm, the way that we write the instructions is called syntax. And the rest of computer programming? That's just practice.

Code-breaking

How good are you at code-breaking? Here is a code crossword. No clues, just the code to break. The key is at the top. All of the letters of the alphabet appear in the key and grid but which letter is which? Work out the code to reveal one of the greatest movie messages of all time. And just to make it a bit more fun, you also have to answer the following questions about the grid:

1 Name a caffeine-loaded programming language.

2 What animal has a crush on you?

3 Name the lump of clay that ran away to have fun.

4 Name an explosive Blondie song?

The answer is posted on the cs4fn website at **www.cs4fn.org**, where there are also some hints about code-breaking.

1	2	3	4	5	6	7	8	9	10	11 A	12	13
14	15 L	16	17	18 R	19	20	21	22	23	24	25	26

■	22	■	11	■	■	11	24	11	■	■	8	■	14	■
12	10	14	7	19	8	18	■	5	16	8	8	7	22	■
■	4	■	14	■	■	8	■	16	■	■	15	10	■	2
11	19	10	3	14	20	■	6	11	13	11	■	19	24	8
■	1	■	11	■	10	18	■	9	■	18	25	8	■	17
24	11 A	18 R	15 L	8	3	■	9	16	22	19	■	22	■	18
■	18	■	■	3	8	■	■	20	■	■	3	■	19	11
21	8	25	22	■	■	22	11	19	16	18	11	19	8	■
■	■	■	■	14	■	11	■	■	■	■	19	10	18	■
■	12	11	1	22	■	15	10	10	12	■	18	10	3	8
■	25	■	11	■	22	8	23	■	■	■	14	■	14	■
22	19	11	22	24	■	22	10	20	21	■	23	■	7	10
■	24	■	19	8	11	3	■	10	■	■	■	■	11	■
26	10	15	8	3	■	11	■	3	11	26	7	8	19	22
■	7	■	9	■	■	7	14	17	■	■	10	■	8	■

A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R
S
T
U
V
W
X
Y
Z

14	7		22	12	11	20	8		7	10	–	10	7	8

20	11	7		24	8	11	18		25	10	16

22	20	18	8	11	3

Cracking Codes

Cryptanalysis, the art of reading secret messages, was invented by Muslim scholars.

The earliest known description of the method needed for this puzzle and used by Walsingham to crack the messages of Mary, Queen of Scots (see page 12) was written by the great Arab philosopher, Al-Kindi in the ninth century.

Meanwhile, somewhere in Canada: A Sodarace story

Sodarace is an online Olympics: a clash of creativity between humans and machines. Who can create the fastest creatures to race over digital terrains? Humans use their ingenuity, drawing and engineering skills to hand craft a creature that can roll, scurry or run as fast as possible over a given 2-dimensional terrain in a Sodarace virtual world. The 'machines' are programs that use a variety of artificial intelligence techniques to similarly create fast creatures.

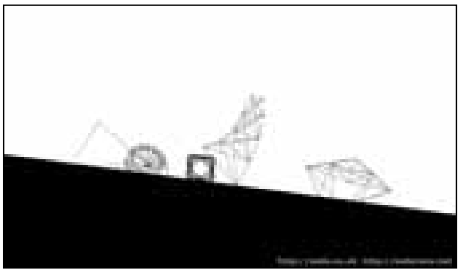
Sodarace players often use what are known as 'genetic algorithms' to create their artificial racers, combining human and machine creativity. Once you've designed the terrain you can take an already developed racer, for example a Daintywalker, and by using a genetic algorithm find the best set of values for the springs and the way they move to get your racer over the course the fastest. This is similar to the process of evolution by natural selection in nature, where animals find the best way to survive in different environments. In the Sodaplanets project Queen Mary students actually developed a whole software package to let you experiment with the sorts of soda lifeforms that would evolve for particular types of planets. You can play with the software that can be downloaded from the Queen Mary SodaPlanet site, create a few planets and see digital evolution for yourself.

Mutations by computer

One of the keys to good survivability in a sodaworld is mutation. Small random changes to the racer often produce racers that are better than any other. When the first ever Sodarace went public, computer scientists spent a great deal of time creating a super daintywalker to cover the racetrack faster than any other previous daintywalkers. They used computer generated mutations to find the best solution, and when they did they posted it in the Sodarace forum (see a video of the race in the webzine) and set to writing a press release to tell the world of their accomplishments the next day.

Meanwhile, somewhere in Canada

That night a kid in Canada found the race, and took up the challenge: to manually try to find a better mutation to beat the Queen Mary racer ... and he did. The next day the scientists found that they had been beaten and had to quickly rewrite the press release. So in the first-ever public Sodarace, human ingenuity, creativity and a strong desire to solve a problem had triumphed over the computer program. News of the human victory went around the news websites of the world. Round one to humanity. Actually, machine intelligence still had some tricks up its digital sleeve ... but that's a different story.



Sodarace the humans vs. machines Olympics is evolving

New software is available for you to play with and give us your comments, and there are some new school activities to try.

TELL YOUR TEACHER!

Help shape the future of Sodarace, it's your project! Go to www.sodarace.net and www.sodaplay.com and have a play

Back (page) to the future.

What does the future hold for computer science? It's always a tricky question to try and answer, and many people in the past got their predictions splendidly wrong. Here are some of the best bloopers (allegedly).

1876

'This "telephone" has too many shortcomings to be seriously considered as a means of communication. The device is inherently of no value to us.' said a Western Union internal memo. Western Union is now one of the USA's largest telecom companies.

1949

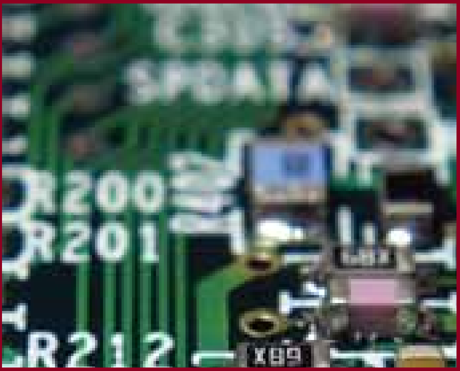
'Where a calculator on the ENIAC is equipped with 18,000 vacuum tubes and weighs 30 tons, computers in the future may have only 1,000 vacuum tubes and weigh only 1.5 tons.' wrote *Popular Mechanics* magazine. If they had been right you'd need a crane with every laptop.

1943

'I think there is a world market for maybe five computers.' said Thomas Watson, chairman of IBM, a company that later went on to revolutionise the home PC market.

1968

'But what ... is it good for?' said an engineer at the Advanced Computing Systems Division of IBM, commenting on the microchip. Now, of course, microchips run all the billions of computing devices on planet Earth.



1977

'There is no reason anyone would want a computer in their home.' said Ken Olson, president, chairman and founder of Digital Equipment Corp. If Ken had been right there would be a lot more table-top room in houses for dusting.

So what are your predictions for the future? Why not email them to us at: cs4fn@dcs.qmul.ac.uk ? We will feature the best in the cs4fn webzine and, who knows, you may even turn out to be right!

> **"There is no reason why anyone would want a computer in their home"**
> **— Ken Olson**

Magic, maths and computer science

Pulling a rabbit out of a hat or making the Statue of Liberty vanish are impressive feats of magic. Magicians are in many ways like computer scientists: a magician must find a method to solve a problem, that problem being, say, making the rabbit appear or the Statue vanish, but without the audience realising how its done. A good magic trick is a combination of method and presentation, in some ways like a computer program: the computer software must have a method to solve the problem (in computer science we call this method, or series of steps, an algorithm), but, unlike magic, software must present the results to the user so they can understand them.

Mathemagic

A mind reading trick with a pocket calculator

Here is a mind reading trick to try with a pocket calculator. Remember that the method (the secret algorithm) and the calculator (the hardware) do the work for you, but you are the one who needs to provide the presentation (you're the 'user interface' here) to make it mysterious and magical.

1 Have someone secretly select a three-digit number and enter it twice into his or her pocket calculator. (For example: 123123) Have them concentrate on the display. You will try to discern their thoughts. (Magic Presentation User Interface needed here!)

2 From across the room (or even over the phone if you want), announce that you predict this number is exactly divisible by 11. Have them verify this by dividing by 11 to find a new whole number with no fractions. Magic!

3 Announce that you feel this new number now on the calculator display is exactly divisible by 13. Have them verify it. More Magic.

4 Now with the number left on the display have them divide by their original three-digit number.

5 Mysteriously announce that the final answer is 7. The Magical Finale.

The secret mathematical algorithm revealed

For your audience hopefully, if you presented magically, this will all look inexplicable, but as a computer scientist you should ask 'But why does this work?' When you look at the mathematics, the answer jumps out. Entering a three-digit number twice (123123) is equivalent to multiplying the three-digit number by 1001 (123 x 1001 = 123123 – try it. It works for any number). Since 1001 = 7 x 11 x 13, their original six-digit number will be divisible by 7, 11, 13, and their originally selected three-digit number.

Understanding how the trick works means that you can come up with your own variations, if rather than have the final prediction come out as 'lucky 7', you want it to be 'unlucky 13' what would you do?

Magic and computer software

It's not surprising that many mathematicians and computer scientists are interested in magic tricks. Working out ways to solve problems, whether predicting a chosen card in a trick or how to reduce the amount of digital data in an MP3 music file without the listener noticing, are very similar. The difference is that computer scientists want to tell the user how it's done. Magicians must keep the method a secret, never revealing it to the audience.

Future Proof

Bill Gates believes CDs and DVDs have had it. It won't be long before the whole back catalogue of music fits on a device in your pocket:

"It's going even faster than we expected…Five years from now people will say 'What's a CD? Why did you have to go to the case and open something up and you couldn't sequence it your own playlist way?' That will be a thing of the past. Even videos in the future will either be on a disk in your pocket or over the Internet, and far more convenient for you."

Bill Gates, Chairman and Chief Software Architect, Microsoft.

Snakey Bites Back

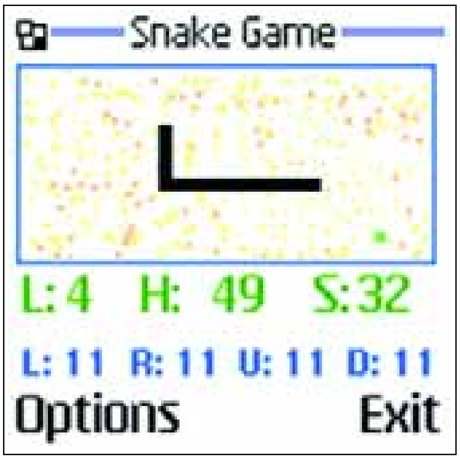
QGames is a fun new way of getting games for your mobile phone. It uses artificial intelligence (AI) techniques know as genetic algorithms to breed a virtual zoo full of evolved versions of the Snake game rated for how difficult a human player will find them. To rank the difficulty levels developer Milan Verma, a student at Queen Mary, University of London built a game playing computer program with ‘human like’ abilities. It 'plays' the game to decide its difficulty. It is based on data from real human game players to give it realistic properties.

The genetic algorithm uses a ‘fitness function’, an approximate way of working out what level of game playing talent the user has. It’s a case of survival of the fittest - as with Darwin’s natural selection that drives evolution in the natural world. Only the best fitting solution for each level of difficulty is selected.

A real user plays a few games so the AI can get a measure of their ability. Milan’s program then sends them a game tailored to their game playing ability. It tailors Snakey for each individual's needs by changing game play factors like speed, snake camouflage, environment and snake mobility. QGames is the first time this idea has been used in a mobile phone.

Milan has bred a zoo full of evolved versions of a game

To learn more about how QGames works and to get two free sample games, visit the Qwacky site linked from **www.cs4fn.org** Updates are available via WAP and GPRS technology.

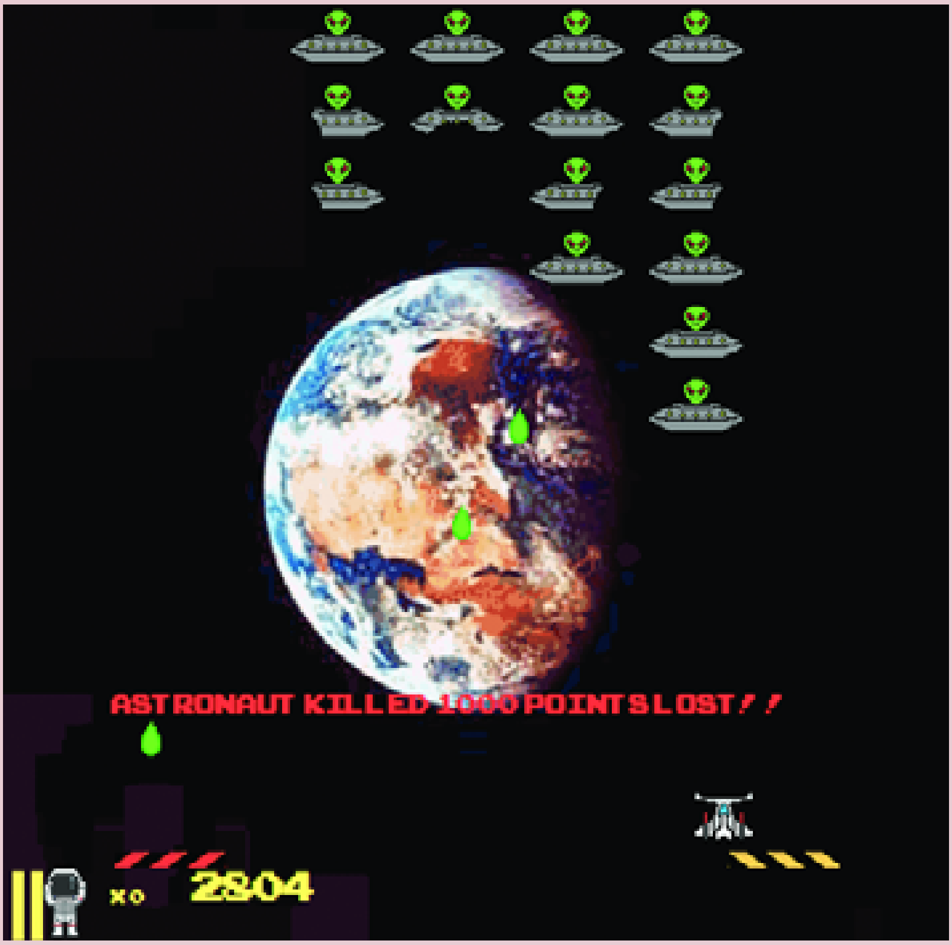


“QGames takes the human versus machine competition straight into your pocket; it’s great fun to play and you never know what trick the AI will try to use to defeat you next time.”

Harry Potter’s Invisibility Cloak

Harry Potter’s invisibility cloak is surely Hogwarts’ magic that science can’t match. Wrap it round you and people just see through you as though you weren’t there. Turns out even that kind of magic can be done with a combination of materials science and computer science. Professor Susumu Tachi of the University of Tokyo has developed a cloak made of thousands of tiny beads. Cameras video what is behind you and a computer system then projects the appropriate image onto the front of the cloak. The beads are made of a special material called retro-reflectrum. It is vital to give the image a natural feel – normal screens give too flat a look, losing the impression of seeing through the person. Now you see me, now you don’t at the flick of a switch.

Pilot Error and Space Invaders



Another plane crashes with all on board killed. The papers blame the pilot. The official report agrees. It was pilot error again. But was the pilot really to blame? Would the pilot have still made the mistake if the cockpit – a very complicated computer interface – had been designed differently? Or is better training the solution? This is something that is important for computer scientists to know – as if it is the design of the human-computer interface that is the problem then they can do something about it. You can help us find out, just by playing a Space Invaders game.

It's a version of the classic Space Invaders game with a human error twist. The program, written by Queen Mary student, Rob Dann, contains opportunities to make errors that lead to you losing all your points: forgetting to reactivate your gun after rescuing an astronaut.

The first results are out. Our Space Invader participants usually avoided making the error as expected, but they did still sometimes make it even with the strong motivation not to. Also players were much more likely to make the mistake if it took them longer than usual to complete an astronaut rescue, suggesting encountering difficulties doing other things can trigger different mistakes.

This all helps us understand how to design other interactive systems in a way that reduces the chance of people making mistakes: vitally important in situations where lives are at stake such as in the way cockpits are designed. We can only design better systems if we understand more about the situations where people are most likely to make mistakes in the first place. *cs4fn* Space Invaders is a fun way to help us find out.

What we need now is for people to play the game ideally all the way to level six...

Interested in helping? Find out more on the *cs4fn* website.

Join the experiment. Play the game.

The Texting Marrakech Game

How to Play

The aim of the texting Marrakech game is to have texted 3 numbers that add up to 15 before the other player. Each player takes it in turn to text the other a number from 1 to 9. Text a number already texted and you immediately lose.

Here's an example game

You text:	**8**
Your opponent texts:	**5**
You text:	**2**
Your opponent texts:	**4**
You text:	**6**
Your opponent texts:	**7**
You text:	**1**

and win
with **8, 6** and **1** = **15**
Notice that if you hadn't texted **6**, the opponent could have done and won with **5,4** and **6** = **15.**

The Secret

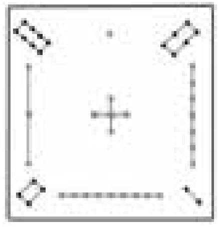
The Marrakech Texting game seems to need you to be amazingly clever at mental maths. In fact all you need is to know a simple bit of mathemagic: a magic square. A magic square is just a square of numbers where each row, column and diagonal add up to the same number.

You can use an ancient magic square known as the Loh Shu to help you win the Marrakech Texting Game. Find out how in the cs4fn webzine: **www.cs4fn.org**

Did you know?

The main town square in Marrakech is called the Magic Square

Magic Squares were invented in China. The earliest one so far discovered is on a scroll from 2800 BC.



Conjuring Cyborg Super Senses:

Exclusive Interview with Cyborg Kevin Warwick

Could humans ever gain super-human powers? Could the blind see with the help of computers? This is the stuff of science fiction comics and films - like Dr Octopus and the X-men or Geordi La Forge from Star Trek who could see using implants in his temples. Can computer science make things like telepathy, dismissed as magic today, become a reality of the future? We talked to the world's first Cyborg, Kevin Warwick about what it is like to have super-human senses.

Our senses are not great compared to other animals. Peregrine falcons can see small prey up to 5 miles away. Polar bears can smell a seal from similar distances. Many animals have senses we do not possess at all, like bats that can sense in ultrasound.

Inventing tools to improve our senses is almost a defining feature of being human. After all, that is all that a pair of glasses does. Computer technology can even let us "see" things other than light. Radio telescopes, for example, let us see distant galaxies and the combination of the Internet and spy satellite technology allows everyone to look down from space on any street in the world.

All of this technology is disconnected though. It doesn't really give us superhuman senses any more than sitting in a tank gives us armour-plated skin. The technology turns the signals into something our limited abilities can sense. We then just use our eyes or ears as normal.

Neuroscientists increasingly understand how our brains and nervous system, the brain's communication system to the rest of the body, works. Signals from our eyes or fingertips pass down our nerves to the brain that turns them into understanding of the world around us. Similarly the brain sends messages out down the nerves to control our movements. Our brains are very adaptable, though. Where the messages come from originally or ultimately go to doesn't matter too much to the brain...and as they are basically just electrical signals computer technology can both detect them and recreate them.

> **"Sitting in a tank does not give you armour plated skin"**

This leads to intriguing questions. What would be possible if we linked computers directly to the nervous system or even the human brain? Can we "cure" disabilities? Could we have super-human senses? Could we sense in ultrasound like a bat and what would it actually feel like? If the computers were linked to the Internet, could it literally take our senses further: outside our bodies altogether? Is any of this even remotely possible?

This all sounds like science fiction or magic but it has already happened. They are the kinds of question Cybernetics Professor, Kevin Warwick of Reading University (www.kevinwarwick.org/) is interested in...and he actually turned himself into the world's first cyborg to find out answers.

A team of neuroscientists implanted a chip directly into the nerves of his arm. It sensed the electrochemical signals travelling along his arm from his brain and transmitted them to a computer. The computer could also send signals to the implant. They then travelled along the nerves of his arm to his brain just like the normal sense signals when he touched something.

By connecting the computer to an ultrasound detector, with the signals from it being sent via the implant to his brain, he could literally feel objects moving towards him even when blindfolded. His brain learnt to interpret the signals from the ultrasound. He really had gained a completely new sense.

The implant was also connected to the Internet. For example, a robot arm copied his arm movements by sensing the signals between his brain and his hand as he flexed his fingers...even though he was in New York and the robotic hand in England.

What does it feel like? According to Kevin:

"Controlling a robot hand on a different continent with my own brain signals was really weird, especially when I could feel how much force the hand was applying. Even after thinking about this for some time I cannot fully imagine the extent of possibilities. Essentially your body can be any physical shape or size whatever that means."

A student made a light-necklace for Kevin's wife, Irena, to wear. When he was relaxed the necklace sensed this via the implant over the Internet and was a cool blue. When he was excited however, his wife would know as her necklace glowed red as a result...even when she was in London and he was still in Reading. When she later had probes inserted into her arm, she didn't need to watch the necklace. She could directly feel the signals from his arm and vice versa, allowing them to communicate directly nervous system to nervous system.

Kevin is excited as to where this could lead:

"One of my heroes is Alexander Graham Bell, so when my wife and I succeeded with the direct nervous system to nervous system communication experiment it was the most exciting thing imaginable. With a blindfold on, my brain received neural pulses that originated from my wife's nervous system. Who knows where this will lead? Hopefully to thought communication, so maybe we won't need speech in the future."

The next step is to place implants not in the arm but directly in the brain. The use of brain implants is actually already quite widespread to help some people with Parkinson's disease live a normal life again. What hasn't been done is to connect brain implants to the Internet and so connect human's brains directly together. That raises even more amazing questions that were previously only the realm of science fiction and pseudoscience. Can more advanced forms of this technology make "telepathy" a reality? Could a human directly sense the thoughts of another, and if so what would it be like?

We could soon know.

Its good to talk:

Alexander Graham Bell

The famous inventor of the telephone Alexander Graham Bell was born in 1847 in Edinburgh, Scotland. His story is a fascinating one, showing that like all great inventions, a combination of talent, timing, drive and a few fortunate mistakes are what's needed to develop a technology that can change the world.

A talented Scot

As a child the young Alexander Graham Bell, Aleck, as he was known to his family, showed remarkable talents. He had the ability to look at the world in a different way, and come up with creative solutions to problems. Aged 14, Bell designed a device to remove the husks from wheat by combining a nailbrush and paddle into a rotary-brushing wheel.

Family talk

The Bell family had a talent with voices. His grandfather had made a name for himself as a notable, but often unemployed, actor. Aleck's mother was deaf, but rather than use her ear trumpet to talk to her like everyone else did, the young Alexander came up with the cunning idea that speaking to her in low, booming tones very close to her forehead would allow her to hear his voice through the vibrations his voice would make. This special bond with his mother gave him a lifelong interest in the education of deaf people, which combined with his inventive genius and some odd twists of fate were to change the world.

A visit to London, and a talking dog

While visiting London with his father, Aleck was fascinated by a demonstration of Sir Charles Wheatstone's "speaking machine", a mechanical contraption that made human like noises. On returning to Edinburgh their father challenged Aleck and his older brother to come up with a machine of their own. After some hard work and scrounging bits from around the place they built a machine with a mouth, throat, nose, movable tongue, and bellow for lungs, and it worked. It made human-like sounds. Delighted by his success Aleck went a step further and massaged the mouth of his Skye terrier so that the dog growls were heard as human words. Pretty ruff on the poor dog.

Speaking of teaching

By the time he was 16, Bell was teaching music and elocution at a boy's boarding school. He was still fascinated by trying to help those with speech problems improve their quality of life, and was very successful in this, later publishing two well-respected books called *The Practical Elocutionist and Stammering* and *Other Impediments of Speech*. Alexander and his brother toured the country giving demonstrations of their techniques to improve peoples' speech. He also started his study at the University of London, where a mistake in reading German was to change his life and lay the foundations for the telecommunications revolution.

See the webzine for the story of how computer science student Lila Harrar was also inspired by a deaf friend... and she has ended up with a commercial product.

A 'silly' German mistake that changed the world

At University, Bell became fascinated by the ideas of German physicist Hermann Von Helmholtz. Von Helmholtz had produced a book, *On The Sensations of Tone*, in which he said that vowel sounds, a, e, i, o and u, could be produced using electrical tuning forks and resonators. However Bell couldn't read German very well, and mistakenly believed that Von Helmholtz had written that vowel sounds could be transmitted over a wire. This misunderstanding changed history. As Bell later stated, "It gave me confidence. If I had been able to read German, I might never have begun my experiments in electricity."

The time for more than dots and dashes

His dreams of transmitting voices over a wire were still spinning round in his creative head. It just needed some new ideas to spark him off again. Samuel Morse had just developed Morse Code and the electronic telegraph, which allowed single messages in the form of long and short electronic pulses, dots and dashes, to be transmitted rapidly along a wire over huge distances. Bell saw the similarities between the idea of being able to send multiple messages and the multiple notes in a musical chord, the "harmonic telegraph" could be a way to send voices.

Chance encounter

Again chance played its role in telecommunications history. At the electrical machine shop of Charles Williams in the USA, Bell ran into young Thomas Watson, a skilled electrical machinist able to build the devices that Bell was devising. The two teamed up and started to work towards making Bell's dream a reality. To make this reality work they needed to invent two things: something to measure a voice at one end, and another device to reproduce the voice at the other, what we would call today the microphone and the speaker.

The speaker accident

June 2, 1875 was a landmark day for team Bell and Watson. Working in their laboratory they were trying to free a reed, a small flat piece of metal, which they had wound too tightly to the pole of an electromagnet. In trying to free it Watson produced a 'twang'. Bell heard the twang and came running. It was a sound similar to the sounds in human speech; this was the solution to producing an electronic voice, a discovery that must have come as a relief for all the dogs in the Boston area.

The mercury microphone

Bell had also discovered that a wire vibrated by his voice while partially dipped in a conducting liquid, like mercury or battery acid, could be made to produce a changing electrical current. They had a device where the voice could be transformed into an electronic signal. Now all that was needed was to put the two inventions together.

The first emergency call

On March 10, 1876, Bell and Watson set out to test their new system. The story goes that Bell knocked over a container with battery acid, which they were using as the conducting liquid in the 'microphone'. Spilled acid tends to be nasty and Bell shouted out "Mr. Watson, come here. I want you!" Watson, working in the next room, heard Bell's cry for help through the wire. The first phone call had been made, and Watson quickly went through to answer it. The telephone was invented, and Bell was only 29 years old.

The world listens

The telephone was finally introduced to the world at the Centennial Exhibition in Philadelphia in 1876. Bell quoted Hamlet over the phone line from the main building 100 yards away, causing the surprised Brazilian Emperor Dom Pedro to exclaim, "My God, it talks", and talk it did. From there on, the rest, as they say, is history. The telephone spread throughout the world changing the way people lived their lives. Though it was not without its social problems. In many upper class homes it was considered to be vulgar. Many people considered it intrusive (just like some people's view of mobile phones today!), but eventually it became indispensable. Bell became rich and famous, and he was only in his mid thirties. The Bell telephone company was set up, and later went on to become AT&T, one of Americas foremost telecommunications giants.

Can't keep a good idea down

Inventor Elisha Gray also independently designed his own version of the telephone. In fact both he and Bell rushed their designs to the US patent office within hours of each other, but Alexander Graham Bell patented his telephone first. With the massive amounts of money to be made Elisha Gray and Alexander Graham Bell entered into a famous legal battle over who had invented the telephone first, and Bell had to fight may legal battles over his lifetime as others claimed they had invented the technology first. In all the legal cases Bell won, partly many claimed because he was such a good communicator and had such a convincing speaking voice. As is often the way few people now remember the other inventors, though different countries now claim the invention of the telephone for different people, so there is plenty to talk about there!

Read Terry Pratchett's brilliant book* Going Postal *for a fun fantasy about inventing and making money from communication technology on DiscWorld.

Bad Wolf...

or a virus in your head?

Lights, cameras, web page

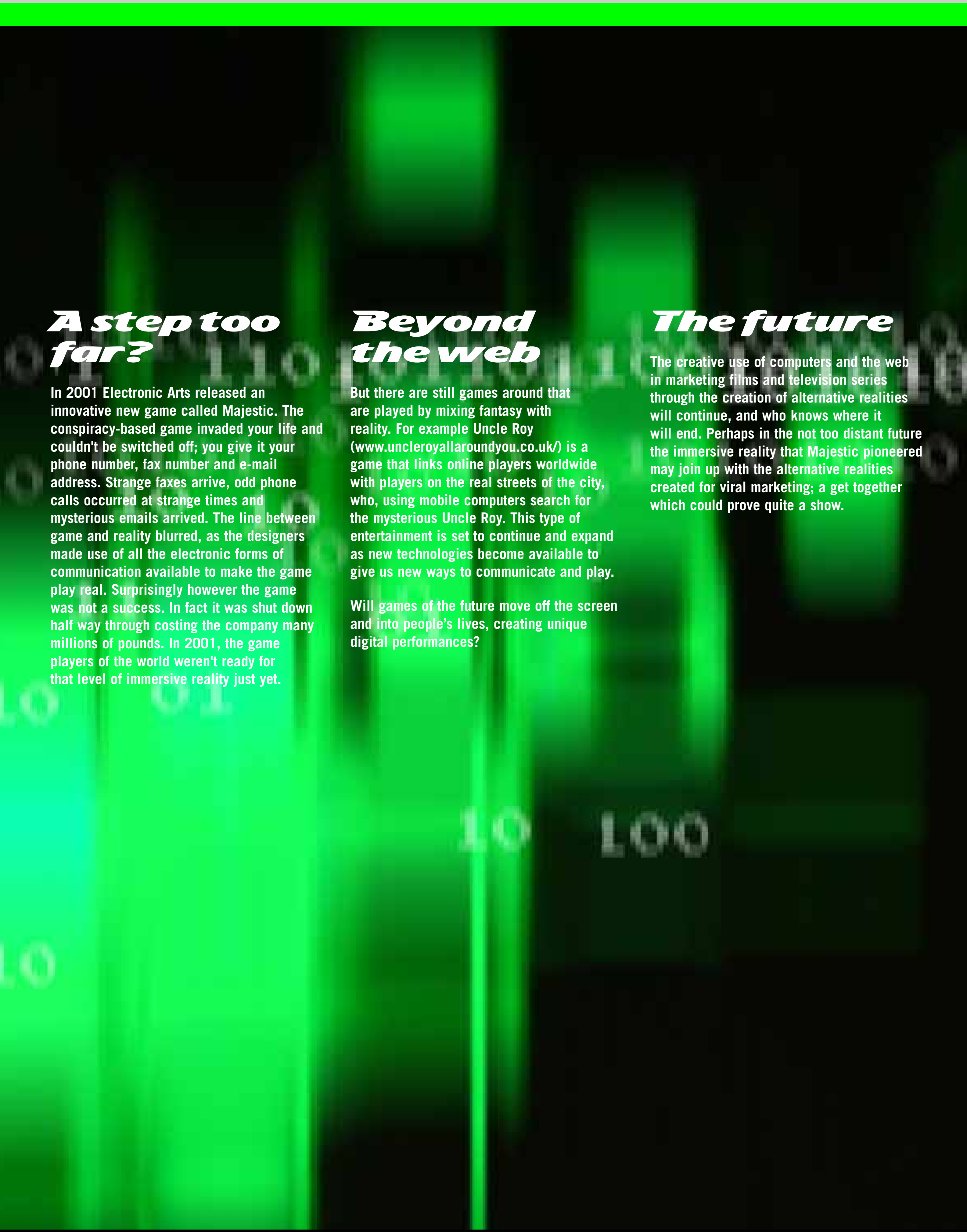
Back in the summer of 2001, the Steven Spielberg film AI: Artificial Intelligence was released. Along with the standard film trailers on the web came rumours about someone named Jeanine Salla. If you searched for the name you found a web page for a university scientist working on advanced robotics and artificial intelligence. It all looked very convincing: lists of scientific papers, a CV, and a full website for Bangalore World University. This was just the start. You could continue the trail and find other sites, drawing you into a world of robotic revolution. But the revolution wasn't about robots. The sites were fakes. The real revolution was the emergence of viral marketing as a tool for marketing movies.

The birth of Viral Marketing

Viral marketing is a different way to raise awareness of your film or TV show. You don't shove it forward using the usual posters or TV campaign. Instead you create a new online reality, and wait; someone will find it, and then they tell their friends, and their friends tell others. They post on bulletin boards and eventually you create an enormous buzz on the Internet. The low budget horror film *The Blair Witch Project* showed for the first time what a cost effective and powerful marketing strategy having a fictional web presence was. It was only a matter of time before this marketing method became mainstream. The fictional sites look as real as anything else on the web; there are layers of secrets and details, hidden text, web sites that look as if they have been 'hacked', with messages hidden in the source code of the web pages. These virtual worlds have the power to draw you in.

BBC's Bad Wolf

A recent example of good viral marketing is in the 2005 BBC TV series *Dr Who*, where clues on the mystery surrounding the identity of the 'Bad Wolf' were laid out across several fictional sites like www.badwolf.org.uk. There were fake sites that were actually referred to in the series (www.whoisdoctorwho.co.uk), a fake site for the UNIT group (www.unit.org.uk) and for the fictional company Geocomtex that appeared in the show (www.geocomtex.net) These sites were not publicised in the early stages. Their presence exploded onto the Internet as fans traded information. The viral marketing helped make the TV series a resounding success, and added a whole new dimension to the series.



A step too far?

In 2001 Electronic Arts released an innovative new game called Majestic. The conspiracy-based game invaded your life and couldn't be switched off; you give it your phone number, fax number and e-mail address. Strange faxes arrive, odd phone calls occurred at strange times and mysterious emails arrived. The line between game and reality blurred, as the designers made use of all the electronic forms of communication available to make the game play real. Surprisingly however the game was not a success. In fact it was shut down half way through costing the company many millions of pounds. In 2001, the game players of the world weren't ready for that level of immersive reality just yet.

Beyond the web

But there are still games around that are played by mixing fantasy with reality. For example Uncle Roy (www.uncleroyallaroundyou.co.uk/) is a game that links online players worldwide with players on the real streets of the city, who, using mobile computers search for the mysterious Uncle Roy. This type of entertainment is set to continue and expand as new technologies become available to give us new ways to communicate and play.

Will games of the future move off the screen and into people's lives, creating unique digital performances?

The future

The creative use of computers and the web in marketing films and television series through the creation of alternative realities will continue, and who knows where it will end. Perhaps in the not too distant future the immersive reality that Majestic pioneered may join up with the alternative realities created for viral marketing; a get together which could prove quite a show.

Magical Memories just shuffling along

"Pick a card, any card!" How often have you heard magicians say that? The normal routine is that you pick a card, the magician shuffles the deck, and abracadabra, reveals your chosen card. But behind this magic often lies some interesting maths, and as we will see later, magicians' shuffles have actually led to the development of new ways for computers to work. Let's start with a trick to amaze your friends.

The 21-card trick.

The Magic effect: Have your friend shuffle a pack of cards and then you deal out single cards left to right into 3 piles of 7 cards, all face up. Your friend has to mentally select one of the cards. They mustn't tell you which card it is, but should tell you the pile it is in. You collect up the cards, and deal them out a card at a time left to right into three piles once more. Again they tell you the pile their card is in, you collect the cards once more, saying you're struggling to "read your friend's mind". Deal the cards out across the table in the three piles again in the same way. Your friend indicates the pile their card is in. Collect the cards again and deal them into the three piles one last time. You immediately announce their card and magically it is in the very middle position of the pack.

How do you perform the trick?

Let's look at the 'mechanics' of the trick. It involves several deals, each apparently shuffling the order of the cards, but doing so in a rather cunning way.

In fact it's really rather simple. All you have to do is make sure you always put the pile your friend selects carefully **between** the other two piles and deal the pack as above. Do that and after the fourth deal the middle card of the middle pile is the chosen card, which you can reveal as you see fit. Try and work out why it works, but then go to the cs4fn web site for an explanation.

Magic and Computers- developing your own algorithms

Once you understand the mechanics you can play with the idea. The order of the chosen pile must not be changed, but the two other piles could, for example, be shuffled before being put together. As long as the chosen pile goes undisturbed between the two other piles of seven cards the order of the other cards doesn't matter. You might want to try and come up with your own additional twists now you know how it's done. The workings of this trick are what's known as an algorithm to computer scientists. The set of steps that you go through to get the trick to work are similar to the way that a computer steps through its instructions in a software program.

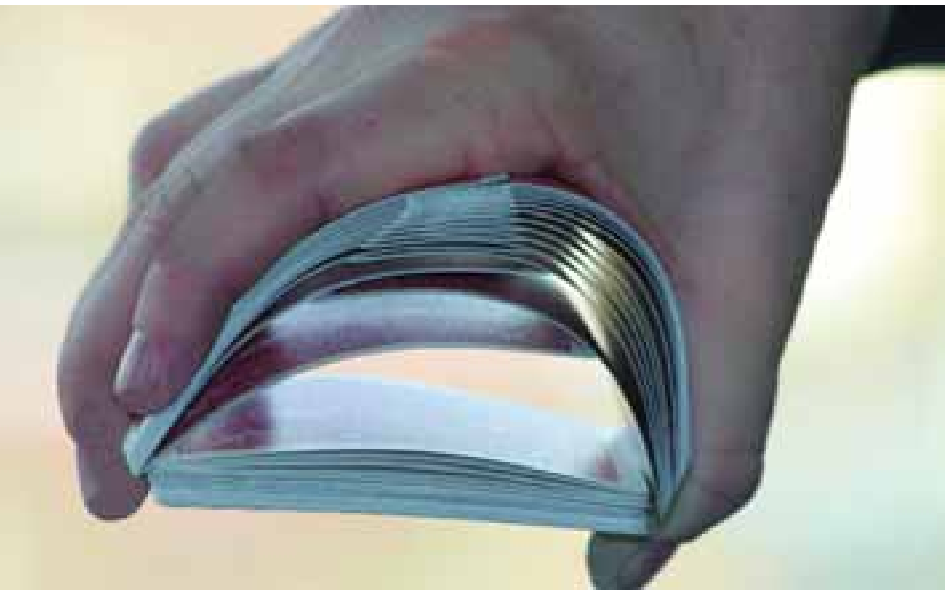
Brent Morris: Magician and Computer Scientist

The magicians' art of shuffling in special ways to make tricks, like the 21 card trick, work can also help us build computers. Magicians want to move cards around efficiently; computers want to move data around in their memory efficiently.

Perfecting the perfect shuffle

In a perfect shuffle, the magician cuts the cards exactly in half and perfectly interlaces them, alternating one card from each half. It takes years of practice to do but does look impressive. There are 2 kinds of perfect shuffle. With an out-shuffle the top card of the deck stays on top. With an in-shuffle the top card moves to the second position of the deck. Magicians know that 8 perfect out-shuffles restore the deck to its original order! It looks like the deck has been really mixed up, but it hasn't.

Computer scientist Brent Morris was fascinated by magic. In particular he became interested in the "perfect shuffle" in high school and has pursued its mathematics for more than 30 years with some amazing results. He earned his Doctorate in Maths from Duke University, and a Master's in Computer Science from John Hopkins University in the United States. He is believed to have the only doctorate in the world in card shuffling. He also holds two US patents on computers designed with shuffles, and has written a book on the subject called *Magic Tricks,*



Card Shuffling, and Dynamic Computer Memories. Why the interest in perfect shuffles?

Binary shifts - as if by magic

You can use perfect shuffles to move the top card to any position in the pack, using a little bit of the maths behind computers: binary. Suppose you want the top card (let's call that position 0) to go to position 6. Write 6 in base 2 (binary), giving 110 (1x4+1x2+0x1). Now read the 0's and 1's from left to right: 1:1:0. Then, working through the 1's and 0's, you perform an out-shuffle for a 0 and an in-shuffle for a 1. In our case that means:

1: an in-shuffle, first

1: another in-shuffle,

0: and finally, an out-shuffle

As if by magic (if you are capable of doing perfect shuffles) the top card will have moved to position 6. Of course it works whatever the number, not just 6.

What does this have to do with the design of computers? You can use exactly the same ideas to move data efficiently around computer memory, which is what Brent Morris discovered and patented.

Curtain call

So as you impress your friends with your 21-card trick, coming up with your own performance ideas and are basking in that applause, remember two things.

Number one: a magician never reveals their secrets.

Number two: computers don't work by Hogwart's magic … only by mathematical magic.

Point, click and Sodarace

Following on from its successful 'run' at the Royal Society summer exhibition Sodarace, the online Olympics competition between human and machine creativity, has taken the next step forward.

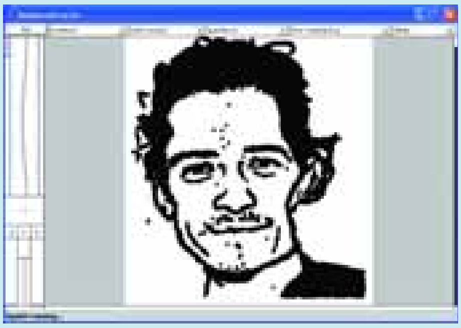
Loads of people throughout the world have used Sodarace for their school and even university projects; from art and design to physics, chemistry, maths and biology, the options seem limitless and it's great fun.

So to help make this even easier you can now download simple to use 'point and click' kiosk software free from **www.sodarace.net** that lets you select the racetrack terrain and the racers for your competitions. It's now straightforward to watch as your selected human and machine created racers go head to head; use your scientific skills to predict which will win on a particular racetrack (or just have a guess).

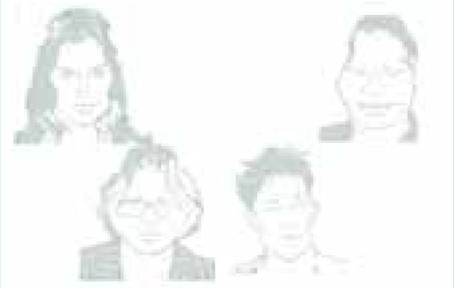
You can also download the first of a range of 'ready to use' lesson plans for classroom activities to try out. Why not ask your teachers to give them a go and let us know how you get on. Keep on racing.

There are 10 kinds of people: those that understand binary and those that don't

Talk of the Toon



Orlando Bloom as a Soda Cartoon



It could be you!

Do you think computers can be creative like humans? They are good at repetitive tasks that are just too boring for humans to do, searching the web for pages on fashion tips or football scores. They can even play some games like chess better than humans. That's just done by searching through more possible moves than a person can. Does that count as creative? It doesn't seem, so. How about drawing? Cartoonists are creative people. If a computer could draw caricatures of people as well as human artists do, would that count as being creative? Queen Mary, University of London, computer science undergraduates, Lila Harrar and Akbar Hussain decided to find out. As part of their course, they created an Artificial Intelligence program based on the BAFTA award winning SodaConstructor, which draws cartoons of people. It works by choosing the most distinctive features of a face and exaggerating them just as human cartoonists do. At the Royal Society Summer Exhibition it was so successful they even had a Robot trying it out. You can get the AI to do a caricature of you (go to the webzine) and find out how it sees you, then decide for yourself whether it is creative or not.

Sonic sleight of hand: The ‘magic’ of MP3

Believe it or not, your MP3 player is actually playing a sonic magic trick on you. MP3 is a format, a way of storing sounds and music so that they take up as little memory in the computer as possible. That way you can have hundreds of your favourite tracks on your iPod, which is really just a small computer disc that stores the digital music. The trick is to find a way to remove some of the sound information without your ears and brain noticing. The computer scientists who developed MP3 are playing a trick on your ears.

Sounds like music

So how is the trick done? Well first we need to look at sound and music. Sound is a change in air pressure. When Madonna sings, her vocal cords change the air pressure and the sound wave passes to our ears. Our eardrum converts this wave of air pressure into a mechanical movement in the tiny bones in our ear. That in turn is changed into a nerve signal that goes to our brain. A microphone works in the same way. The pressure wave moves a part of the microphone called the diaphragm, and it's this movement that turns the sound pressure wave into an electronic signal. When we are recording digital music we take many millions of very rapid samples – measurements – of the electrical signal and turn them into numbers to store.

Frequency asked questions

One of the things that characterises the sound wave is the frequency it is made of. Frequency means a regular repeating pattern. A very dull sound might just be a single frequency. The sound gets louder then softer, say every second, and repeats this cycle every second until you switch it off. This tedious sound would have a frequency of 1 Hertz (1 Hertz means one 'cycle' per second). Normal music doesn't sound at all like this dull repeating noise, but as it happens you can take any sound or music and, using some special mathematics, convert it into a set of different frequencies. Each of these frequencies alone just sound dull but added together it makes the music. The same idea is used in a music synthesiser; you can make the sound of any instrument by adding together the right frequencies. So we now know that our music can be described as frequencies, and we can start to play tricks.

Ear, ear

Turns out that our ears, though very clever, don't do everything well. Certain frequencies will stop you hearing other close by frequencies. These special frequencies 'mask' the presence of the others nearby. Once you know this (it was discovered through lots of experiments on hearing), then you know there is no point in using up valuable computer memory storing information on the frequencies your ears can't hear. So you don't store them. Kazam! You remove these frequencies altogether, but because they were masked by the other frequencies you can hear, you don't notice they have vanished.

Bunny in the headlights

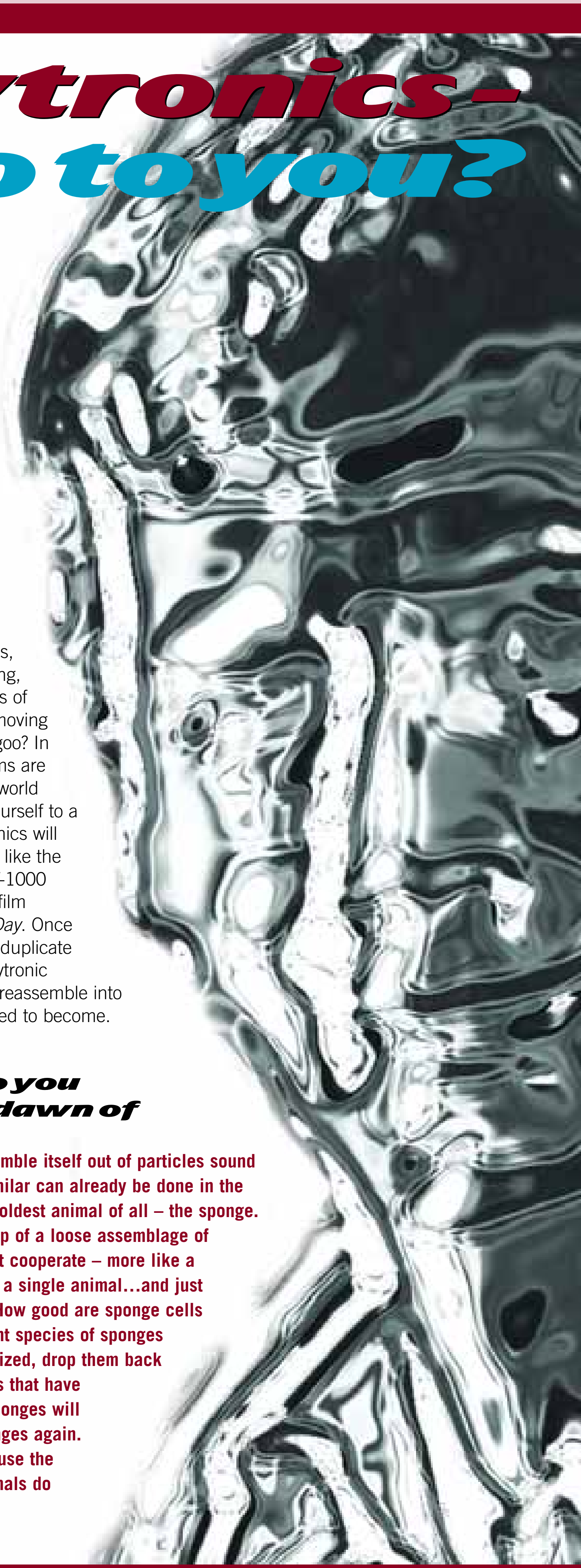
There is another cunning effect you can use. When we look at a bright light, for a while afterwards, we can’t see a dim light. Our eyes change so we aren't dazzled by the bright flash, but this leaves our ability to see dimmer lights reduced for some time. The same thing happens with sounds. A loud sound will stop you hearing a quieter sound that follows just after. So we can look at the digital music signal and work out all the places where a loud sound is followed by a quiet sound, and then cut the quiet sound out. This saves computer memory, and again the trick is that your ears won't notice the quieter sound is missing.

Sonic sleight of hand

It's using these tricks to vanish those parts of the sounds your ears won't notice are missing that gives MP3 its great ability to compress music. We need to store less data but the music sounds just as good. To be able to give the world MP3, Computer Scientists needed to work on ways to do the maths, as well as understand how our ears work. So the next time you listen to your MP3 player just think of all the maths and computing making up the wonderful din.

Yeh but no but... Read about compressing Vicky Pollard in the webzine

Claytronics – from goo to you?



It would be magical to be able to teleport: Star Trek’s transporter was a clever device that allowed the characters to teleport from their spaceship to the planet. Devised by Gene Rodenberry, Star Trek's creator, as a way to avoid the costly special effects of landing spaceships in the TV series, the transporter became the key to many an episode.

In effect, the transporter scans the passenger’s atoms, disintegrating them into energy, then transmitting the energy to recreate a copy of that "pattern of atoms” on the planet below. It's a nice idea but the physics is frightening, disintegrating atoms and turning them into energy is exactly what a nuclear bomb does, and a nuclear bomb only converts a tiny fraction of matter to energy. If we really want to be able to create copies of ourselves at a distance we need to think of less explosive methods.

The physics is frightening ... exactly what a nuclear bomb does

It is just this problem that computer scientists in the USA at the Carnegie-Mellon University Synthetic Reality Project are looking at. Their solution: a new science called Claytronics. By using tiny programmable machines they hope to develop 'programmable matter', where millions of tiny devices called "claytronic atoms" or "catoms" would assemble into the shape of any object you want, connecting and disconnecting as they move. Currently they have developed experimental catoms that connect and move via magnets. These early devices are around four centimetres in size, much larger than the size the team want to develop, but you have to start somewhere.

The challenges are both in the technology, how do you build these tiny catoms, but also in the programming, how do you instruct billions of little machines to build a moving copy of yourself from the goo? In the future, if these problems are solved, we can imagine a world where you can transmit yourself to a meeting where the claytronics will build a copy of you, much like the smart liquid metal of the T-1000 Terminator android in the film *Terminator 2 Judgement Day*. Once your meeting is over, your duplicate will melt away and the claytronic programmable matter will reassemble into whatever it is next instructed to become.

From goo to you ...since the dawn of time

Does a robot that can assemble itself out of particles sound far-fetched? Something similar can already be done in the animal world...and by the oldest animal of all – the sponge. A sponge’s body is made up of a loose assemblage of separate types of cells that cooperate – more like a colony of cells rather than a single animal...and just like the claytronics idea. How good are sponge cells at assembling? Put different species of sponges in a liquidizer. Once liquidized, drop them back into seawater and the cells that have survived of the different sponges will reassemble back into sponges again. Computer Scientists often use the inspiration of the way animals do things to create advanced technology.

tlhIngan Hol Dajatlh'a' (Do you speak Klingon)

Language is something we take for granted. We learn it as a child, maybe study a new language at school, or pick up some choice phrases for a holiday trip. A language consists of words - the bits it is made of - and grammar - the way these bits are put together. "The cat sat on the mat" makes sense. “Cat mat on sat the” doesn't. We have the same words but the rules of grammar are broken ... the information is lost. This confused list of words still follows a rule: the words are arranged alphabetically. But without the right rules, the rules of grammar, the words can't do what they need to do, which is let us know where that pesky cat is.

When we have to communicate with a computer and give it instructions we want to make life easy for ourselves. In the beginning programmers were forced to use binary – lists of 1's and 0's to tell the computer what to do. That's because computers are really just a very complex box of switches, and the 1's and 0's told the computer which parts of its circuits to switch on and off, but it made writing the programs a nightmare. It was asking you to speak in an alien tongue.

A better way to communicate with our computers

Something had to change, so those long-suffering programmers looked to what they knew best, human languages, to find a solution. In the same way as a word is made up from letters you could start to think of lists of binary instructions that you could associate together into a simple command to which you could attach some meaningful name. So rather than tell the computer in 1's and 0's ('0101010' say) to put a particular set of values into a particular circuit, you could use the phrase 'LET X=2' to mean the same. The computer understands how to turn a number into binary. It knows that 2 is 10 in binary (that means take a two and add no ones to it, just like a normal decimal 10 means take a ten and add no ones), so this useful idea of a computer language that WE understand means we don't need to talk the nasty alien computer lingo. The language instruction tells the computer to find a place in its memory, which the programmer wants to call X and to put 2 in there.

So to programming languages

The programmer's language is translated to the language the computer uses by following a list of instructions. In the same way as with a human language, if we get the grammar wrong and say typed '=LET 2 X' the computer wouldn't know what to do, the information is there but the grammar is wonky. So computer languages developed following many of the rules of human languages to try and keep it as easy as possible for human programmers to program. BASIC, Beginners All Symbolic Instruction Code was an early attempt. That's why, for example, there are university courses on Computer Science and Linguistics. The two are very closely related.

The Universal Translator?

There are now computer programs that translate human languages. You may have seen these on the Google or Babel Fish web sites for example. What's happening here is that, to translate say French text into German, the French is first translated to a language the computer understands and can work with. The text in that computer language is then translated back into German ... so it's passed through that strange alien binary language of the computer somewhere in between. Of course its very basic, Google just uses a simple approach to translation, but there are many researchers around the world trying to develop software that takes into account the complex rules of human languages and grammar. Some systems even try to learn languages from scratch like a child, in effect trying to produce the universal translator similar to the idea in Star Trek, which can translate any language to English. (Of course it would be easier to use the TARDIS telepathic circuits like Dr Who ... a small summer project there for anyone interested:-)

Synthetic Languages

We tend to think of human languages as always having been around in their current form, and computer languages as new, but human languages need to have developed from somewhere, and that's another fascinating story where computers have helped. Some people have even created their own spoken languages such as Esperanto. The basic rules and words of Esperanto were proposed by LL Zamenhof at the end of the 19th century. The idea was to create a world language that everyone could speak. It never really took off, though there are still many people who learn it and use it.

Which brings us to Klingon, a made up language from Star Trek developed by Marc Okrand, (see the Klingon Language Institute at www.kli.org/). It has words, a fixed grammar and there are lots of 'native speakers' out there, so you guessed it ... there are programs that translate English to Klingon. It's interesting to think that a software program to translate a made up alien language does it by using its own made up alien language in the computer. As Mr Spock would say 'Quite Fascinating!'

Qapla' ("Success!")

Footnote

In the interests of Galactic Peace it should also be pointed out that there are also programs that translate English into Vulcan, Romulan, Ewok, Wookie, Ferengi ...

Traditionally machine translation has involved professional human linguists manually writing lots of translation rules for the machines to follow. Recently there have been great advances in what is known as statistical machine translation where the machine learns the translations rules automatically.

It does this using a 'parallel corpus': just lots of pairs of sentences; one a sentence in the original language, the other its translation. Parallel corpora are extracted from multi-lingual news sources like the BBC web site where professional human translators have done the translations.

بغداد ١-١ (اف ب) ذكرت وكالة الانباء العراقية الرسمية ان نائب
رئيس مجلس قيادة الثورة في العراق عزة ابراهيم استقبل اليوم
الاربعاء في بغداد رئيس مجلس ادارة المركز السعودي لتطوير
الصادرات عبد الرحمن الزامل.

Let's look at an example translation:

Above is some original arabic with its translation below:

Machine Translation: Baghdad 1-1 (AFP) – The official Iraqi news agency reported that the Chinese vice-president of the Revolutionary Command Council in Iraq, Izzat Ibrahim, met today in Baghdad, chairman of the Saudi Export Development Center, Abdel Rahman al-Zamil.

Human Translation: Baghdad 1-1 (AFP) – Iraq's official news agency reported that the Deputy Chairman of the Iraqi Revolutionary Command Council, Izzet Ibrahim, today met with Abdul Rahman al-Zamil, Managing Director of the Saudi Center for Export Development.

This example shows a sentence from an Arabic newspaper then its translation by the Queen Mary, University of London's statistical machine translator, and finally a translation by a professional human translator. The statistical translation does allow a reader to get a rough understanding of the original Arabic sentence. There are several mistakes, though. Mistranslating the "Managing Director" of the export development center as its "chairman" is perhaps not too much of a problem. Mistranslating "Deputy Chairman" as the "Chinese vice-president" is very bad. That kind of mistranslation could easily lead to problems. That reminds me of the point in *The Hitch-Hiker's Guide to the Galaxy* where Arthur Dent's words "I seem to be having tremendous difficulty with my lifestyle," slip down a wormhole in space-time to be overheard by the Vl'hurg commander across a conference table. Unfortunately this was understood in the Vl'hurg tongue as the most dreadful insult imaginable, resulting in them waging terrible war for centuries....

For now the human's are still the best translators but the machines are learning from them fast!

Strictly X-Factor

TV talent shows like *X-Factor*, or *Soapstar Superstars* have always been popular. Its not just the talent on show that make them must see TV – it's having the right mix of personalities in the judges too. Simon Cowell has made a career of being rude – even reaching the dizzy heights of a guest appearance on *The Simpsons*. In contrast judge Sharon Osborne's on screen persona is far more supportive. It's often the tension between the judges that makes good TV.

However, if you believe Dr Who, the future of game shows will be robot judges like AnneDroid in the space age version of *The Weakest Link*…let's look at the robot future. How might you go about designing computer judges?

We need to write a program. We don't want to have to describe new judges from scratch each time. We want to do as little as possible to describe each new one.

What makes a judge?

First let's describe a basic judge. We will create a plan, a bit like an architect's plan of a building. It can then be used to build individual judges. What's the X-factor that makes a judge a judge? First we need to decide on some characteristics of judges. We can make a list of them. The only thing common to all judges is they have different personalities and they make judgements on people. Let's simply say a judge's personality can be either supportive or rude, and their judgements are just marks out of 10 for whoever they are watching.

```
Character : SUPPORTIVE OR RUDE.
Judgement : 1 TO 10.
```

So let's start to specify (describe) Judges as people with a personality and capable of thinking of a mark.

```
DESCRIPTION OF a Judge:
    Character personality.
    Judgement mark.
```

All we are saying here is whenever we create a Judge it will have a personal character (it will be either RUDE or SUPPORTIVE). For any given judge we will refer to their character as "personality". It will also have a current judgement, which we will refer to as "mark": a number between 1 and 10.

Best Behaviour

We are now able to say whether a judge is rude or supportive, but we haven't actually said what that means. We need to set out the behaviours associated with being rude and supportive. To keep it simple, let us say that the personality shows in the things they say. A rude judge will say "You're a disgrace" unless they are awarding a mark above 8/10. For high marks they will grudgingly say "You were ok I suppose".

```
TO Speak:
        IF (personality IS Rude) AND
            (mark <= 8)
        THEN SAY "You're a disgrace".
        IF (personality IS Rude) AND
            (mark > 8)
        THEN SAY "You were ok I suppose".
```

It would be easy for us to give them lots more things to choose to say in a similar way. We can do the same for a supportive judge. They will say "You were stunning" if they award more than 5 out of 10 and otherwise say "You tried hard".

Ten out of Ten

The other thing that judges do is actually come up with their judgement. We will assume, to keep it simple here, that they just think of a random number – essentially throw a 10 sided dice under the desk with numbers 1-10 on. Judges' decisions can sometimes look like that on TV!

```
TO MakeJudgement:
        mark = RANDOM (1 TO 10).
```

Putting that all together to make our full judge description we get:

Our final plan for making judges

```
DESCRIPTION OF A Judge:
    Character personality.
    Judgement mark.

    TO Speak:
        IF (personality IS Rude) AND
            (mark <= 8)
        THEN SAY "You're a disgrace".

        IF (personality IS Rude) AND
            (mark > 8)
        THEN SAY "You were ok I suppose".

        IF (personality IS Supportive) AND
            (mark > 5)
        THEN SAY "You were stunning".

        IF (personality IS Supportive) AND
            (mark <= 5)
        THEN SAY "You tried hard".

    TO MakeJudgement:
        mark = RANDOM (1 TO 10).
```

Kind words for our contestants?

Suppose now we want to create a rude judge, called SimonCoward. We can use the plan. We need to say what its personality is (Judges just think of a mark when they actually see an act so we don't have to give a mark now.)

```
SimonCoward IS A NEW Judge WITH
                    personality Rude.
```

This creates a new judge called SimonCoward and makes it Rude. We could similarly create a rude AnneDroid:

```
AnneDroid IS A NEW Judge WITH
                    personality Rude.
```

For a supportive judge that we decide to call SharONN we would just say:

```
SharONN IS A NEW Judge WITH
                    personality Supportive.
```

Whereas in the specification we are describing a plan to use to create a Judge, here we are actually using that plan and making different Judges. So this way we can quickly and easily make new judge clones without copying out all the description again.

A classless society?

Computer Scientists are lazy beings – if they can find a way to do something that involves less work, they do it, allowing them to stay in bed longer. The idea we have been using to save work here is just that of describing classes of things and their properties and behaviour. Scientists do that a lot:

Birds have feathers (a property) and lay eggs (a behaviour).

Spiders have eight legs (a property) and make silk (a behaviour)

We can say something is a particular instance of a class of thing and that tells us a lot about it without having to spell it all out each time (even for fictional ones): eg

Hedwig is a bird. (so feathers and eggs)

Charlotte is a spider. (so legs and silk)

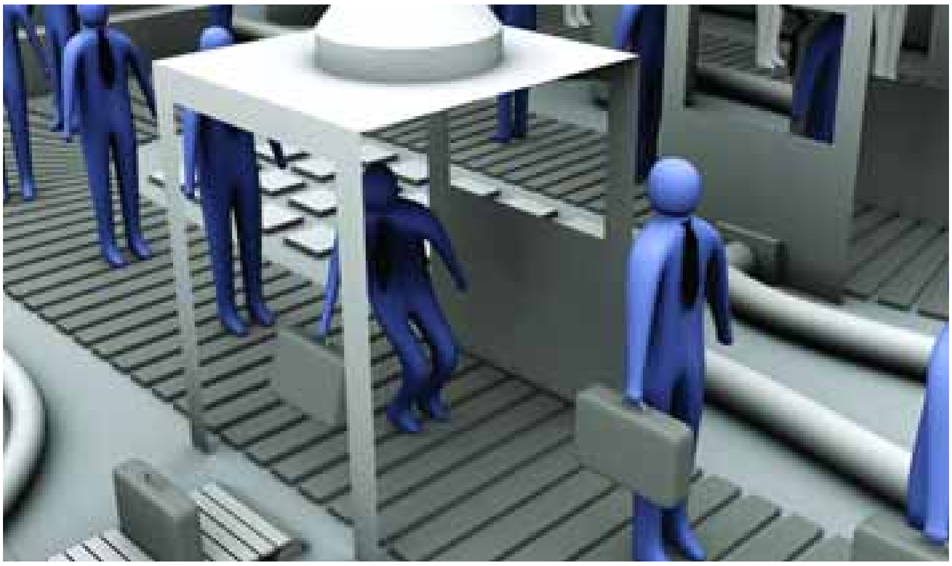
So we can now create judges to our hearts content, fixing their personalities and putting the words in their mouths based on our single description of what a Judge is.

All Change

We have specified what it means to be a robotic judge and we've only had to specify the basics of Judgeness once to do it. That means that if we decide to change anything in the basic judge (like giving them a better way to come up with a mark than randomly or having them choose things to say from a big database of supportive or rude comments) changing it in the plan will apply to all the judges of whatever kind.

What we have created is our first object-based program – it would be relatively easy to convert this into a program in a programming language like Java or C#.

We could create robot performers in a similar way (after all don't all the winners seem to merge into one in the end?). We would then also have to write some instructions about how to work out who won – does the audience have a vote? When do judges make judgements? When can they speak their mind? How many get knocked out each week? That's no harder. Why not give it a try and judge for yourself?



So how does a SharONN Judge have daughter KeLEE judges without any help from Ozzie? See **www.cs4fn.org**

Back (page) to the drawing board.

You may have heard the term software engineering. Building a complex computer program is like building any complex machine or structure. It takes skill, professionalism, teamwork and the ability to learn from your mistakes. Every type of engineering throughout history has had its share of disasters. Early steam engines exploded, the Titanic famously sank, the Hindenburg airship caught fire, and the Tay Bridge in Scotland fell down.

Software engineering is no different. Here are a few of its disasters to learn from (more in the cs4fn webzine):

The Ariane 5 Rocket (problems with big numbers in small spaces, 1996)

In 1996 an Ariane 5 rocket exploded forty seconds after lift-off. The project had taken 10 years, and cost $500 million. This spectacular software failure was due to squeezing a big number into the computer memory reserved for a small one. There wasn't enough space to hold the rocket's speed when it was passed to another smaller memory store. This caused the rocket to veer off course, break up and explode.

The moral: Make sure numbers fit their destinations.

Mars Climate Orbiter (a weighty problem in space, 1999)

Programmers work in teams to build software for a space mission. Unfortunately for NASA's $125 million Mars Climate Orbiter, two teams didn't know what the other was up to. One team was using Imperial weight measures (pounds). The other was using metric (kilograms). The mistake wasn't found until, when finally in space, the two programs spoke to each other, got very confused and caused the spacecraft to become lost in space.

The moral: software engineering is also about team communication

AT&T Crash (The day the phones stopped working…all of them, 1990)

In late 1989, AT&T engineers upgraded the software of their 114 US switching centres: the computers that make the connections so your phone links to the one you are calling. Each computer was duplicated so if one went wrong the other would take over. On January 15th 1990, they stopped working: 70 million calls failed. The problem was in a single line of code out of millions…and it was in both computers' copies. AT&T lost $1 billion as customers fled to their competitors.

The moral: with software, duplication doesn't always help

Big numbers can hurt

USS Yorktown Stops (a big something caused by nothing, 1998)

Dividing by zero is a bad idea. The answer doesn't exist. However a crewmember of the computer controlled guided-missile cruiser USS Yorktown mistakenly entered a zero on their console. It resulted in the computer program trying to do an impossible divide by zero. The program crashed and caused a failure in other linked computer systems on the ship, eventually shutting down the ship's engines, leaving it drifting for hours.

The moral: always check numbers are as expected.

The Pentium Chip Error (not enough numbers in the table, 1994)

The Pentium Chip used a look up table (LUT) to do division; basically it uses a pre-calculated set of numbers to speed things up. The LUT should have contained 1066 elements, but when the numbers were downloaded a bug in the software only put in 1061 of them. No one checked, and the chip went to manufacture with those numbers missing. When the mistake was found the chips all had to be replaced. The cost was more than $4 billion.

The moral: keep testing all the way.

Dividing by Zero is a bad idea

Bugs can kill

The missing American Patriot Missile (a problem with bad timekeeping, 1991)

In 1991, during the first Gulf War, an American Patriot Missile battery in Dharan, Saudi Arabia, failed to shoot down an incoming Iraqi Scud missile. The Scud missile hit an army barracks with many casualties. State-of-the-art computers controlled the Patriot missile, but there was a problem. To work Patriot needed to accurately know the time. This was done with an internal clock that started to tick when the computer was first switched on. However the computer program, when converting the internal clock time into the time used by the guidance system introduced a tiny mistake. It rounded the number down slightly. With each passing second the error became larger until finally it was enough to make the missile miss.

The moral: small mistakes in calculations often build into big mistakes.

And Finally expect the unexpected

In the early days of electronic computers they used relays, electromechanical switches that rocked up and down to switch the electrical circuits. Grace Murray Hopper, who was in charge of the team working with the Mark II computer, (an early electro-mechanical device), found that a moth had flown through the window and blocked one of the relays, so shutting the system down. This is arguably where the term computer 'bug' comes from.

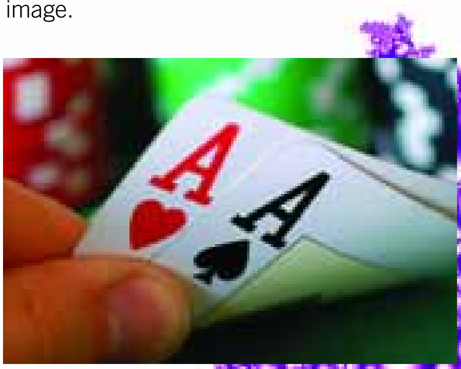
The moral: The things some moths get into can be shocking!

The Fractal Casino Royale

Casino Royale: a new Bond and a new title sequence. Gone are the silhouettes of naked women of all the previous films. After all it would hardly be appropriate. Bond falls in love. But what replaces them? A poker theme for the gambling addicted modern day? Well yes: it is a film about a card game after all. Look more closely though and you will see it's computer science that's replaced the women: fractal imagery.

Look carefully at the clubs as they expand in the title sequence. Each leaf buds off a new smaller club, which then does the same again, creating an ever more intricate pattern. It is a fractal image: an image that is self-similar on smaller and smaller scales. It turns out that natural processes such as the way trees and ferns grow can be modelled mathematically in the same way - break off the frond of a fern and it looks very much like the original only smaller.

That means that fractals are a very good way to quickly create realistic computer-generated images of plants. Fractals have also been suggested as a rival to jpeg for compressing images, though it never really took off. Known as fractal compression, the idea was to look for fractal self-similarity in images and then store the rules for creating the fractals rather than all the detail of the original image.



Fractal images are very easy to generate using a process called recursion. It's a way of problem solving (and programming) where a problem is broken into smaller versions of the same problem. These smaller but similar problems can then be solved in the same way. Eventually the problem is broken into a problem so small and trivial the answer is obvious.

The self-similar nature of the ever smaller problems is the same as the self-similar nature of the ever smaller fractal images. That means the rules to generate fractal images are very similar to the computer programs that solve problems using recursion.

Interested in generating your own fractal images? You can do it using GeomLab. Go to the webzine to find out more.

Coupled cups create communication

Let's raise a glass to toast the computer scientists who have developed the 'connected cup'. Researchers at MIT in the States have developed drinking glasses that are linked by computer. The cups are covered with sensors that detect when the cup is moved, tilted or when you take a sip from it. The cups also glow; they have different colours produced by small light emitting diodes built into them, and they also have a small motor in them to make them shake. But the clever twist is that the cups are connected by a wireless radio link. This means that when a distant friend drinks your health you will know, as your cup will glow as they take a sip. It also means that you can share a toast with friends from different parts of the planet. Other applications of this new technology also include being able to monitor the elderly to ensure they are drinking enough water, or a mother at home can cause their children's cup at school to glow and shake when its break time just to show how much they care.

It's just one of the fascinating research projects worldwide looking at how the connectedness of the Internet combined with computers built into the things we use can change our lives. It's something to think about when you're supping your next glass of water.

Music to your ears

Making music has always been a social thing. People play in bands or orchestras, and even composers often work in pairs like the Beatle's Lennon and McCartney, Rolling Stones Mick Jagger and Keith Richards or Neil Tennant and Chris Lowe of the Pet Shop Boys.

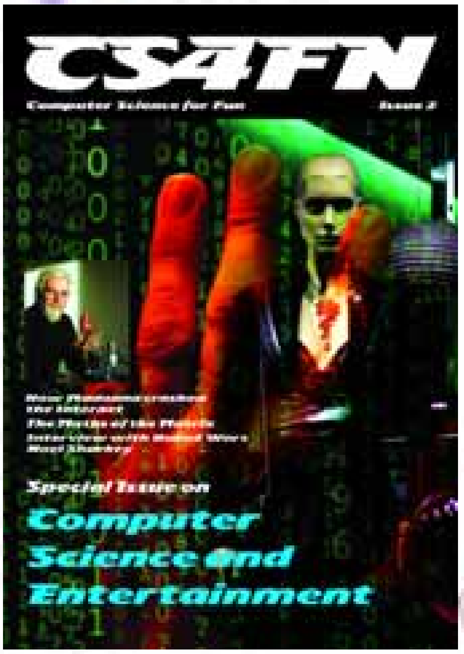
Up till now playing or composing in groups has tended to need the people concerned to be in the same place at the same time. That used to be the case for people working together as well. The Internet and mobile phones have changed all that though. Networked games allow people to play together without ever meeting too. So if we can both work and play together in groups across continents, what about making music? What kind of instruments would allow people to work creatively and compose together, and what kind of music would you get?

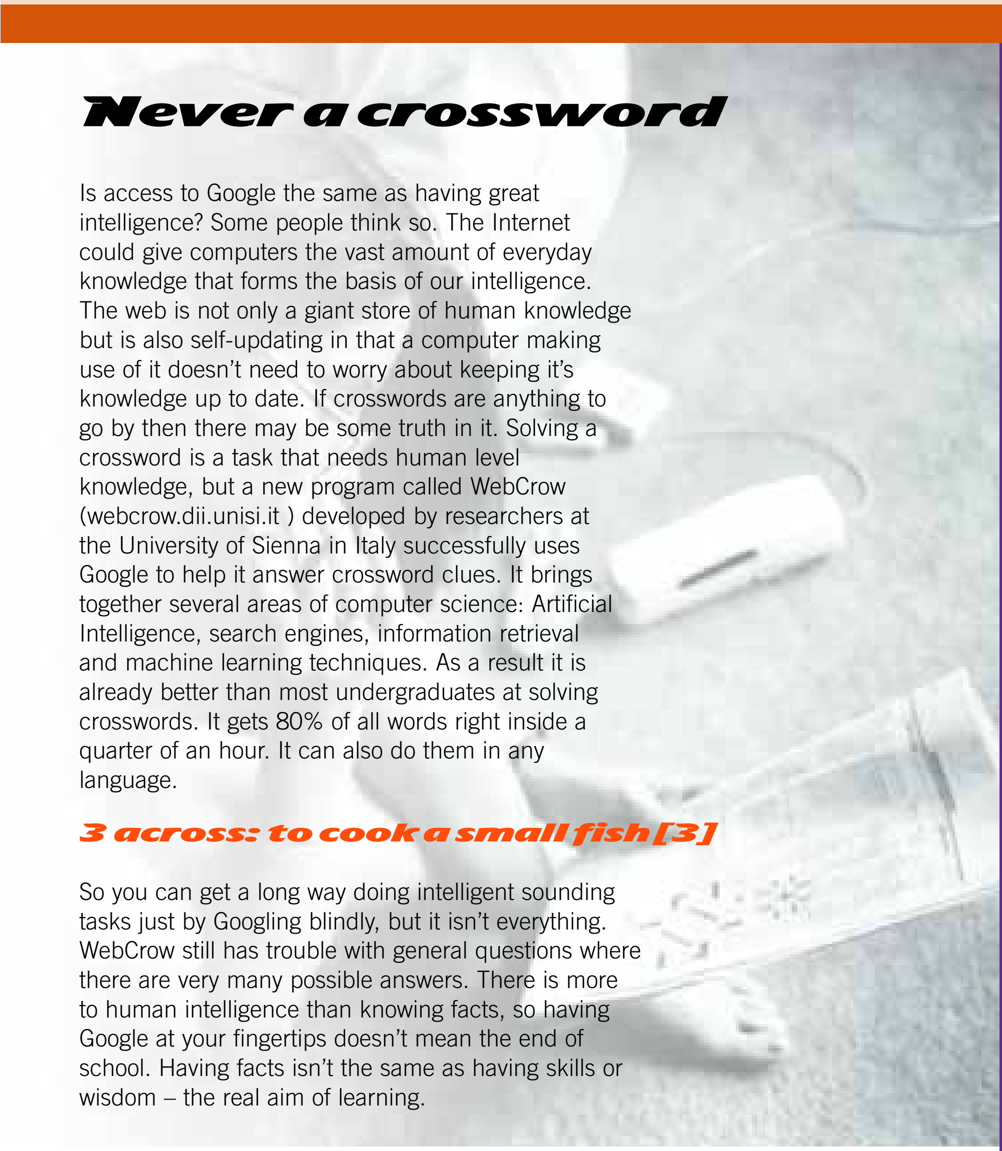
Daisyphone, is a new way of finding out that you can take part in. Its floral dot-to-dot design lets people play loops of music together, continually adapting the loop depending on the sound and on what others do. You can all see and hear what the others taking part do, wherever in the world they are. It is also very visual – you can hear the patterns you doodle, or even what your name sounds like. It was developed as part of a research project at Queen Mary that is also exploring the collaborations of jazz musicians jamming and how their creative intensity can be captured online, but it is freely available for anyone to use.

So give it a go, learn to play a new group instrument, exploring the emerging world of online group creativity and maybe make some new musical friends at the same time.

Visit : www.cs4fn.org

You can hear the patterns you jointly doodle





Never a crossword

Is access to Google the same as having great intelligence? Some people think so. The Internet could give computers the vast amount of everyday knowledge that forms the basis of our intelligence. The web is not only a giant store of human knowledge but is also self-updating in that a computer making use of it doesn't need to worry about keeping it's knowledge up to date. If crosswords are anything to go by then there may be some truth in it. Solving a crossword is a task that needs human level knowledge, but a new program called WebCrow (webcrow.dii.unisi.it) developed by researchers at the University of Sienna in Italy successfully uses Google to help it answer crossword clues. It brings together several areas of computer science: Artificial Intelligence, search engines, information retrieval and machine learning techniques. As a result it is already better than most undergraduates at solving crosswords. It gets 80% of all words right inside a quarter of an hour. It can also do them in any language.

3 across: to cook a small fish [3]

So you can get a long way doing intelligent sounding tasks just by Googling blindly, but it isn't everything. WebCrow still has trouble with general questions where there are very many possible answers. There is more to human intelligence than knowing facts, so having Google at your fingertips doesn't mean the end of school. Having facts isn't the same as having skills or wisdom – the real aim of learning.

Crosswords helped the war effort

Intelligence is a tricky thing to tie down, though crosswords have a history of being used to indicate intelligence. In World War II British 'Intelligence' even used the ability to do the Times crossword as a recruitment tool – for the people they needed to help crack the German codes working with the very first computers.

A funny thing happened on the way to the computer

Laugh and the world laughs with you they say, but what if you're a computer. Can a computer have a 'sense of humour'? Try our joke 'Turing Test':

Can you tell which joke is written by a human and which by a computer program (called JAPE) Have your vote on the cs4fn website.

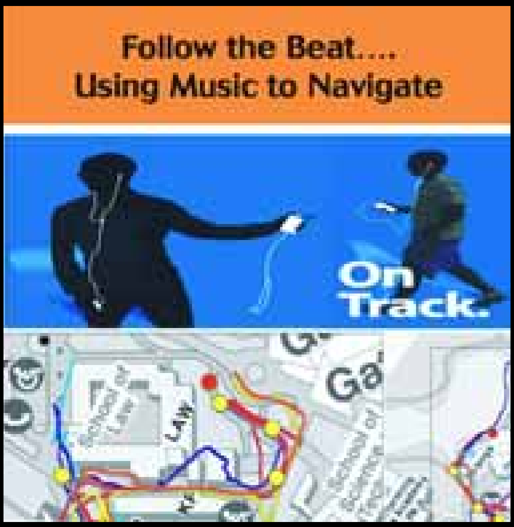
Lacking a sense of humour? WebCrow has trouble with puns too ... perhaps it needs to be introduced to JAPE, the joke telling computer (see box)

On Track: Follow the Beat

Female crickets have long since navigated to their partners using their songs. Now you can do it too. The OnTrack system developed at the Universities of Swansea and Waikato in New Zealand combines global satellite positioning with iPod music players to tell you where to go to the sound of your own music.

The volume is increased or decreased to tell you if you are moving towards or away from the target of your affection. The music is sent to the left or right headphone to guide you to the left or right.

So join the fun, join the crickets and find your music mate.



Hogwarts magic?

Dragons, floating candles, talking portraits, house elves, Hippogriffs and even werewolves, such are the magical and mythical contents of Hogwarts school. The Harry Potter movies have been a worldwide success. Amazing when you think they all started in the fertile imagination of author JK Rowling as she sat in a café in Edinburgh. So how have the wonderful magical creations written on the page been able to spring to life to amaze Harry, Ron and Hermione. Muggles without magical talent turn to the computer to make the pages come alive, and with today's astonishing computer graphics it's possible to turn magic and imagination into film reality.

Computer graphics imagery, or CGI, is at the heart of many of today's most popular movies, from Harry Potter to Spiderman, from Mission Impossible to the Matrix and beyond. Sometimes they create the impossible like the Hippogriff. Sometimes they allow actors to do impossible things by replacing the real actor with a digital copy, called a synthespian.

CGI for movies often makes use of cutting edge computer science research. For example the genesis wave in Star Trek 2: The Wrath of Khan was the first time so called particle systems were used: a method of modelling natural phenomena like fire by hundreds of interacting parts. The watery aliens in James Cameron's the Abyss were produced using new computer modelling methods to create realistic water shapes, and the animal stampede in The Lion King cartoon, made use of models of herding behaviours in real animals to make it frighteningly realistic. Often the computer scientists working on these films use their results in scientific papers, which they present at scientific conferences. What we see on the screen as movie magic today is frequently in fact the state of the art in computer science research. More recent films like the Pirates of the Caribbean: The Curse of the Black Pearl, The Chronicles of Narnia: The Lion, the Witch and the Wardrobe, King Kong, and War of the Worlds would not have been the smash hits they were without their stunning computer graphics.

The importance of research into new and better CGI has been part and parcel of the success of filmmakers like George Lucas, Steven Spielberg and Peter Jackson. For example, George Lucas set up his own computer graphics research laboratory called Pixar to ensure his films can benefit from new and previously unseen special effects on screen. Most of the people working in this company have PhD's in computer science, and it's their creativity and hard work along with those like them who help put the Magic into Hogwarts.

The maths of the Matrix

In the hit movies series 'The Matrix' the heroes and villains undertake spectacular fight sequences, on the ground, in the air and even on busy motorways. The film called on some state-of-the-art special effects to take the movements of actors and seamlessly put them into the picture or replace them with computer-generated doubles. At the heart of these effects is the ability to 'capture' human motion. That is to be able to exactly record how the actors' body, face and hands move. With this information special effects wizards can apply the motions to computer generated 'puppets', such as the loathsome Golem in the Lord of the Rings films, King Kong or even Clone Troopers in Star Wars. With the advent of new ways to capture actors movements, quicker and better ways to produce increasingly realistic computer graphics, and super fast computer hardware to do all the calculations, film making today is limited only by the filmmakers' imagination and budget.

How do motion capture systems work? These systems need to be able to find the positioning of parts of the actors body accurately as they move round, and the principle they often use is as old as the hills. In days of yore when people wanted to find the distance to an object, such as a ship at sea or the target for a cannon ball, they would use trigonometry. Yes it's all done with school maths: sines, cosines and tangents. Suppose you had two lookouts a known distance apart on two hilltops. If each lookout could see the 'target' and measure the angle to that target, then some simple trigonometry, the law of sines, would give you the distance. This process was known as triangulation, as the two lookouts and the target form the three corners of a triangle, and the laws of triangle geometry give you the answer you need.

Surveyors still use the same method to make the accurate maps used for in-car navigation systems. Satnav systems work out positions from satellites – 21st century navigation only exists thanks to simple trig.

Jump forward to motion capture

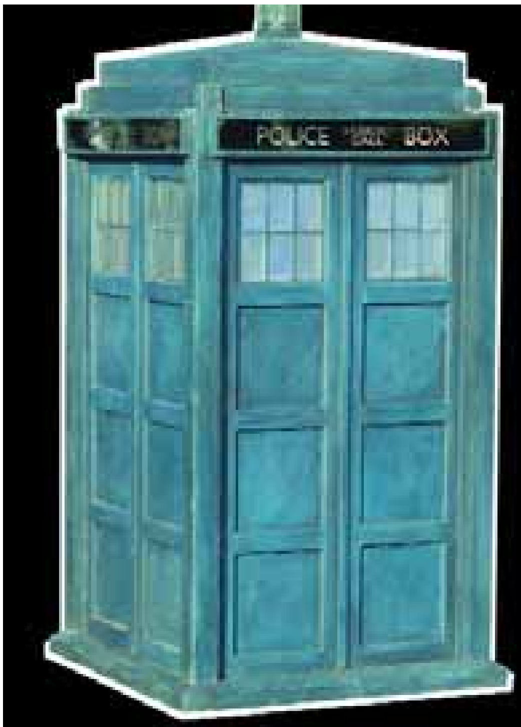
You may have seen 'behind the scenes' DVD extras on 'The making of...' movies where the actors jump around wearing lycra bodysuits covered with small reflective bobbles on stage. This isn't a fashion statement, its just triangulation for the 21st century (see the 'Maths of the Matrix). The 'bobbles' on the suits provide 'landmarks.' They are just something obvious to look for. Around the studio are multiple cameras that can track the position of the bobbles. Cameras work in pairs (sometimes more) to calculate the distance by triangulation to the body landmarks they can see. As the actors move they recalculate the distances to producing a track of how that landmark has moved over time.

So what happens when a camera looses track of a landmark? For example the actor might turn away. The answer is simple. There are enough cameras looking into the area that at least two will always be able to see each of the bobbles wherever they are. It's like having ten or twenty lookouts stationed around the place, each keeping an eye on a different part of the studio.

Clearly there's a lot of calculations to do, hundreds of triangulations per second using some clever maths and geometry. Getting accurate motion from, for example, people's faces means even more markers need to be added (normally with a special gum that sticks them to the actors face!). The computer software is able to put together all the data from the different triangulations to measure the actor's movements, which is normally displayed on a 'stick man' graphic so the director can see what's been captured. Once the motion sequence is in the computers it can be used to operate a virtual graphics puppet (called an avatar) to replace the actor or allow the avatar to interact with other graphics generated characters in a scene in a movie or computer game.

Motion capture can also be used in sports, to see how well athletes are performing, in medicine to detect problems in how people walk or move, and in engineering to see how people move for example in a new design for a car. There is also work underway to use motion capture to try and understand how people interact with one another in group conversations and meetings, so psychologists can benefit from using this 'Movie magic' too.

What's been described above is called optical motion capture, as the cameras 'see' the landmarks, but some motion capture systems work with magnetic markers, some work with sound, using acoustic signals like a sonar to find the distances, and some even strap the poor actors into a mechanical 'rig' to measure how their body moves. Whatever technology is used the software does the schoolkid sums to produce the motion sequence to let the movie magic happen. Quite moving in its simplicity really.



Werewolf Bobble Hats

To animate the werewolf in the Dr Who episode "Tooth and Claw" the actor had to wear a hat containing a bobble on a stick to mark the top of his head, as the werewolf was so much taller than the actor was. It also helped the other actors look in the right place and meant the actor knew when to duck as he went through doors.

If you love Dr Who, you might be interested in www.visittorchwood.co.uk/... the Ice Cream sounds yummy!

Swat a way to drive

Flies are small, fast and rather cunning. Try to swat one and you will see just how efficient their brain is, even though it has so few brain cells that each one of them can be counted and given a number. A fly's brain is a wonderful proof that, if you know what you're doing, you can efficiently perform clever calculations with a minimum of hardware. The average household fly's ability to detect movement in the surrounding environment, whether it's a fly swat or your hand, is due to some cunning wiring in their brain.

Speedy calculations

Movement is measured by detecting something changing position over time. The ratio distance/time gives us the speed, and flies have built in speed detectors. In the fly's eye, a wonderful piece of optical engineering in itself with hundreds of lenses forming the mosaic of the compound eye, each lens looks at a different part of the surrounding world, and so each registers if something is at a particular position in space.

All the lenses are also linked by a series of nerve cells. These nerve cells each have a different delay. That means a signal takes longer to pass along one nerve than another. When a lens spots an object in its part of the world, say position A, this causes a signal to fire into the nerve cells, and these signals spread out with different delays to the other lenses' positions.

The separation between the different areas that the lenses view (distance) and the delays in the connecting nerve cells (time) are such that a whole range of possible speeds are coded in the nerve cells. The fly's brain just has to match the speed of the passing object with one of the speeds that are encoded in the nerve cells. When the object moves from A to B, the fly knows the correct speed if the first delayed signal from position A arrives at the same time as the new signal at position B. The arrival of the two signals is correlated. That means they are linked by a well-defined relation, in this case the speed they are representing.

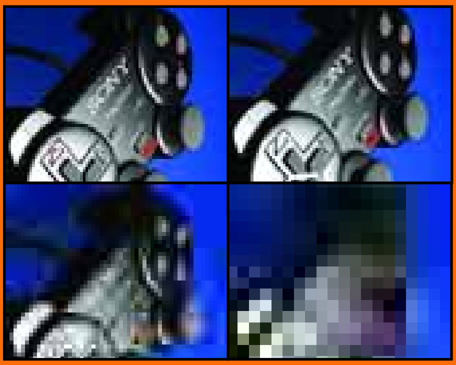
Do Locusts like Star Wars?

Understanding the way that insects see gives us clever new ways to build things, and can also lead to some bizarre experiments. Researchers in Newcastle showed locusts edited highlights from the original movie Star Wars. Why you might ask? Do locusts enjoy a good Science Fiction movie? It turns out that the researchers were looking to see if locusts could detect collisions. There are plenty of those in the battles between X-wing fighters and Tie fighters. They also wanted to know if this collision detecting ability could be turned into a design for a computer chip. The work, part-funded by car-maker Volvo, used such a strange way to examine locust's vision that it won an Ig Nobel award in 2005. Ig Noble awards are presented each year for weird and wonderful scientific experiments, and have the motto 'Research that makes people laugh then think'. You can find out more at http://improbable.com

Car Crash – who is to blame?

So what happens when we start to use these insect 'eye' detectors in cars, building smart cars with the artificial intelligence (AI) taking over from the driver to avoid hitting other things? If we do build cars with fly or locust like intelligence, which avoid accidents like flies avoid swatting or can spot possible collisions like locusts, an interesting question arises. Suppose an accident does happen. Who's to blame? Is it the car driver – are they in charge of the vehicle? Is it the AI to blame? Who is responsible for that: the car manufacturers? Is it the computer scientists who wrote the program? What will insurance companies decide? As computer science makes new things possible society will need to decide how to deal with them. Unlike the smart cars, these decisions aren't something we can avoid.

Picture This? JPEG It!



Streaming big brother

The Big Brother TV show is frighteningly popular, the chance to watch the housemates 24/7 as they do their tasks, bicker and even …sleep. While the housemates are at the mercy of TV producer, Big Brother, behind the scenes the viewer has a new freedom to decide when or where they watch the show. This ability to send video to mobile phones and over the Internet has changed the way people want to watch TV. As the next generation of video mobile phones and the networks capable of sending the signals come on line we will see more TV shows becoming mobile 24/7 digital events. Behind this new TV technology there is some clever computer science. Video signals are big; they have lots of data, both moving pictures and sound. So scientists have had to come up with some clever ways to cut down on the amount of data they send without you noticing. If you want to know how this sneaky trick is done you can read all about it overleaf. For starters, a simpler problem is to learn some tricks with stills (see below). Computer scientists have also had to make sure that popular web broadcasts like Big Brother evictions don’t overwhelm the Internet. It wasn’t always so. You can read how singer Madonna ‘killed’ the Internet a few years ago on page 48. Whatever TV producers come up with in the future, as your enjoy the show remember its computer science that’s really behind the scenes pulling the strings.

Looking at a picture from your digital camera or a digital movie, it’s all just 11001100011– hardly inspiring, and I don’t really see what it means!

The human brain is thought to have around half its volume given over to making sense of vision. A surprising fact perhaps, but it just goes to show how hard understanding the world we see around us is. Scientists the world over are interested in vision. We can try to understand it by looking at the biology of the brain. We can do experiments to try and measure how we go from the image in our eyes to being able to understand what we look at. Computer scientists can also try to build machines that can ‘see’ to give insight into the way human’s do it. If half your brain is needed to see then you can be sure that some fairly hefty calculations are going on in your ‘little grey cells’ and its making use of lots and lots of information.

Information, or data, is something that computer scientists respect. The amount of data needed to accomplish a task determines the amount of calculation needed, and calculations cost, both in the time taken and in the hardware used. The brain obviously does it pretty well. So when computer scientists looked at the problem of making a movie or TV show take up the least possible space on your computer, or of using the least possible amount of data to be transmitted, it’s not surprising that they looked to their brains for help.

See it the psychologists way

Psychologists had discovered that human observers are very sensitive to changes in the amount of light in an image (called the luminance), but less so to the changes in colour. This is because our eyes (which turn the light waves from what we are looking at into nerve signals on the retina at the back of your eyes) have two sets of detectors. One is for measuring the amount of light and a separate set help measure the colour of the light. It turns out there are less colour detectors. So when we look at what data we can remove from the image, represented as a stream of ones and zeros, we choose changes in colour. If we make this reduction, by putting in less colour information, our brains don’t miss it. Meddle with the luminance and we pick it up easily.

We can throw out some colour and our brains don’t notice, but psychologists tell us there are some other things we can remove too. We often hear that people don’t bother to read the ‘small print’ in contracts, or that a ‘small detail’ was easily overlooked. Well our brains do the same with everything we see. Our brains can’t read the ‘small print’ in images. We can take any image and through some clever maths turn it into a ‘top ten’ of detail. At number one is the pattern of big changes of light over the image, and way down the list are the pattern of how smaller changes in light affect the image. This ‘top ten’ is called the spatial frequency spectrum of the image. It tells us what patterns at different levels of detail add together to make the original. So with this knowledge we can decide that our image only needs say the top five, and remove the other lower chart (spectrum) entries. Turns out that again our brains won’t miss the data. We don’t notice it much, so like colour some levels of detail can be reduced.

Leave it Out!

This ‘removing things we won’t notice’ idea is what makes JPEG images work. We can reduce the data for an image by reducing the way we calculate colour changes and changes in level of detail. We can apply these ideas to little blocks of the images. So we take the whole image and break it into bits, and we cut down the data in each bit using our understanding of the brain. What we end up with is not an image but a set of instructions on how to build the image. We send the instructions and when the computer receives them it uses a ‘codex’, a small program that knows how to turn the instructions for each block into a picture, to recreate the original (well not quite the original but our brains are sufficiently fooled). We can take this removal to the extreme if we want really small amounts of data, or high data compression, but eventually our brains will notice. So it’s about understanding what level of removal our brains won’t miss and fixing the minimum amount of data the computer wants to handle. As always it’s a trade off, but this trade off is smart.

So the next time you’re looking at a digital image think how JPEG is playing tricks on you to create the illusion. What you see is all just 11001100011. The same tricks and more are played when you watch a movie. Go to the webzine to find out more.

MPEG: Movie magic

What's your favourite film? We call films "films" because of what they are made of. Moving Pictures revolutionized entertainment and the secret was in the way sequences of pictures could be printed onto rolls of film. The future of movies and TV isn't film, it's streams of digits. Both are going digital... So what's your favourite MPEG then?

Let's get moving

Photos went digital using something called JPEG (see 45). It is just an agreed way of converting single pictures into streams of 1s and 0s. It plays some magic tricks on our eyes. Once you understand how the JPEG magic trick is done it's easy to see how movies might go digital. They are just a stream of still images: frames that are each different from the other. So a movie could just be a series of JPEG images. It turns out we can do better than the obvious though.

Movies are tougher than photos because there is so much data - so many images just for a few seconds. When we show them in quick succession, one after the other, another of our brain's little tricks called 'perseverance of vision', combines and blends the individual frames all into a continuous motion. We 'see the movie'. So how can we trick our brains again and not have to send every single frame (or block) of a movie one by one? The answer comes from watching lots and lots of movies!

All change please

If it's the same from frame to frame, why bother sending it over and over again?

Watch a movie or TV show. Things move on the screen but just as importantly some things stay the same. The blocks making the tree in the background or the studio set are the same from scene to scene. So if it's the same from frame to frame, why bother sending it over and over again. This is the idea behind MPEG, the movie format (MPEG stands for Moving Pictures Experts Group); send only the block of data you need. So we have to send some frames, and of course we use JPEG to do this as it's already reduced the data in each block by mind tricks. But we look at the movie first and compute which frames (set of blocks) we need to send in a complete (well JPEG) form and which frames actually just contain bits of other frames. So we send instructions on how to use other frames to build these. If the tree is static, send it only once, and then send instructions that in other frames just add in the tree we already have.

The frames we send in full are called I frames (I for Intra frame). The frames that are built from I frames, by moving blocks around are called P frames (P is for predicted). P frames must follow I frames in the movie, as they are built from I frame data. We send the information on how to move the blocks only, but if it turns out that something new appears in the frame, a ball appears or a door opens, then we need to send the instructions for the new block and how to create it. Since we are only sending movement instructions to shift existing blocks around or having to create the occasional new block, the data is much less than that we would need if we had to send everything.

The story so far

So far, we have built I frames from compressing a set of images in the original movie. We have used the information in I frames to build P frames by sending instructions for shifting the I frame data around. Both have reduced the amount of data we had to send. Is there anything else we can do to make the data sent even smaller? Yes there is. We have sent the data for I frames and P frames, but in between we can create B frames. B frames (Bi-directional frames) are the cheapest to create from a data point of view. We build B frames by taking an I frame and a later P frame (built from the I frame data), and use them both to make up what's going on in between. A B frame takes the information in the I and P frames on either side of it and looks at how to use the information to create the picture it is supposed to be. It takes blocks from the I frame and P frame and moves them around as is required for the action to move seamlessly between them. Only at the last will a B frame need to contain any new information. So in order of reducing data we have the I frame JPEG tricks first, then less data using the block move around to get a P frame and finally the B frame which has the least as it makes use of the data in the other two. The amount of data used for a movie can be changed by selecting the proportions of the different types of frames. MPEG uses clever computation based on the fact that movies tend to contain sequences of frames that don't change much, and it uses this to drop the amount of data needed.

Order, Order

I, P and B frames are instructions not pictures. So first, for a movie, we need to process all the frames to find the best way to crush the data needed down. A P frame follows in time after an I frame. It's using the I frame data, but a B frame in between needs both the I and the P data to work. This means that the order that the frames are sent isn't the same as the order the viewer sees them in. All the I and P frames need to go first then the intermediate B frame instructions follow. So when you are watching an MPEG movie, or digital TV which uses similar techniques, your computer is actually doing some time travel. It's storing the I and P frames till it gets the instructions for B frames and then slots the newly created B frames in between the I and P's to show to you.

The next time you're watching a film think how MPEG is using tricks in both space and time to create the illusion of movement. What you see is all just 11001100011. Oh but don"t forget to enjoy the movie.

If it's the same from frame to frame, why bother sending it over and over again?

Die another Day? Or How Madonna crashed the Internet

When pop star Madonna took to the stage at Brixton Academy in 2001 for a rare appearance she made Internet history and caused more that a little Internet misery. Her concert performance was webcast; that is it was broadcast real time over the Internet. A record-breaking audience of 9 million tuned in, and that's where the trouble started...

The Internet's early career

The Internet started its career as a way of sending text messages between military bases. What was important was that the message got through, even if parts of the network were damaged say, during times of war. The vision was to build a communications system that could not fail; even if individual computers did, the Internet would never crash. The text messages were split up into tiny packets of information and each of these was sent with an address and their position in the message over the wire. Going via a series of computer links it reached its destination a bit like someone sending a car home bit by bit through the post and then rebuilding it. Because it's split up the different bits can go by different routes.

Express yourself (but be polite please)

To send all these bits of information a set of protocols (ways of communicating between the computers making up the Internet) were devised. When passing on a packet of information the sending machine first asks the receiving machine if it is both there and ready. If it replies yes then the packet is sent. Then, being a polite protocol, the sender asks the receiver if the packets all arrived safely. This way, with the right address, the packets can find the best way to go from A to B. If on the way some of the links in the chain are damaged and don't reply, the messages can be sent by a different route. Similarly if some of the packets gets lost in transit between links and need to be resent, or packets are delayed in being sent because they have to go by a round about route, the protocol can work round it. It's just a matter of time before all the packets arrive at the final destination and can be put back in order. With text the time taken to get there doesn't really matter that much.

The Internet gets into the groove

The problem with live pop videos, like a Madonna concert, is that it's no use if the last part of the song arrives first, or you have to wait half an hour for the middle chorus to turn up, or the last word in a sentence vanishes. It needs to all arrival in real time. After all, that is how it's being sung. So to make web casting work there needs to be something different, a new way of sending the packets. It needs to be fast and it needs to deal with lots more packets as video images carry a gigantic amount of data. The solution is to add something new to the Internet, called an overlay network. This sits on top of the normal wiring but behaves very differently.

The Internet turns rock and roll rebel

So the new real time transmission protocol gets a bit rock and roll, and stops being quite so polite. It takes the packets and throws them quickly onto the Internet. If the receiver catches them, fine. If it doesn't, then so what? The sender is too busy to check like in the old days. It has to keep up with the music! If the packets are kept small, an odd one lost won't be missed. This overlay network called the Mbone, lets people tune into the transmissions like a TV station. All these packages are being thrown around and if you want to you can join in and pick them up.

It's like someone sending a car bit by bit through the post... different bits can go by different routes

Crazy for you

The Madonna webcast was one of the first real tests of this new type of approach. She had millions of eager fans, but it was early days for the technology. Most people watching had slow dial-up modems rather than broadband. Also the number of computers making up the links in the Internet were small and of limited power. As more and more people tuned in to watch, more and more packets needed to be sent and more and more of the links started to clog up. Like dozens of cars all racing to get through a tunnel there were traffic jams. Packets that couldn't get through tried to find other routes to their destination … which also ended up blocked. If they did finally arrive they couldn't get through onto the viewers PC as the connection was slow, and if they did, very many were too late to be of any use. It was Internet gridlock.

> **Like dozens of cars all racing to get through a tunnel there were traffic jams. It was Internet gridlock.**

Who's that girl?

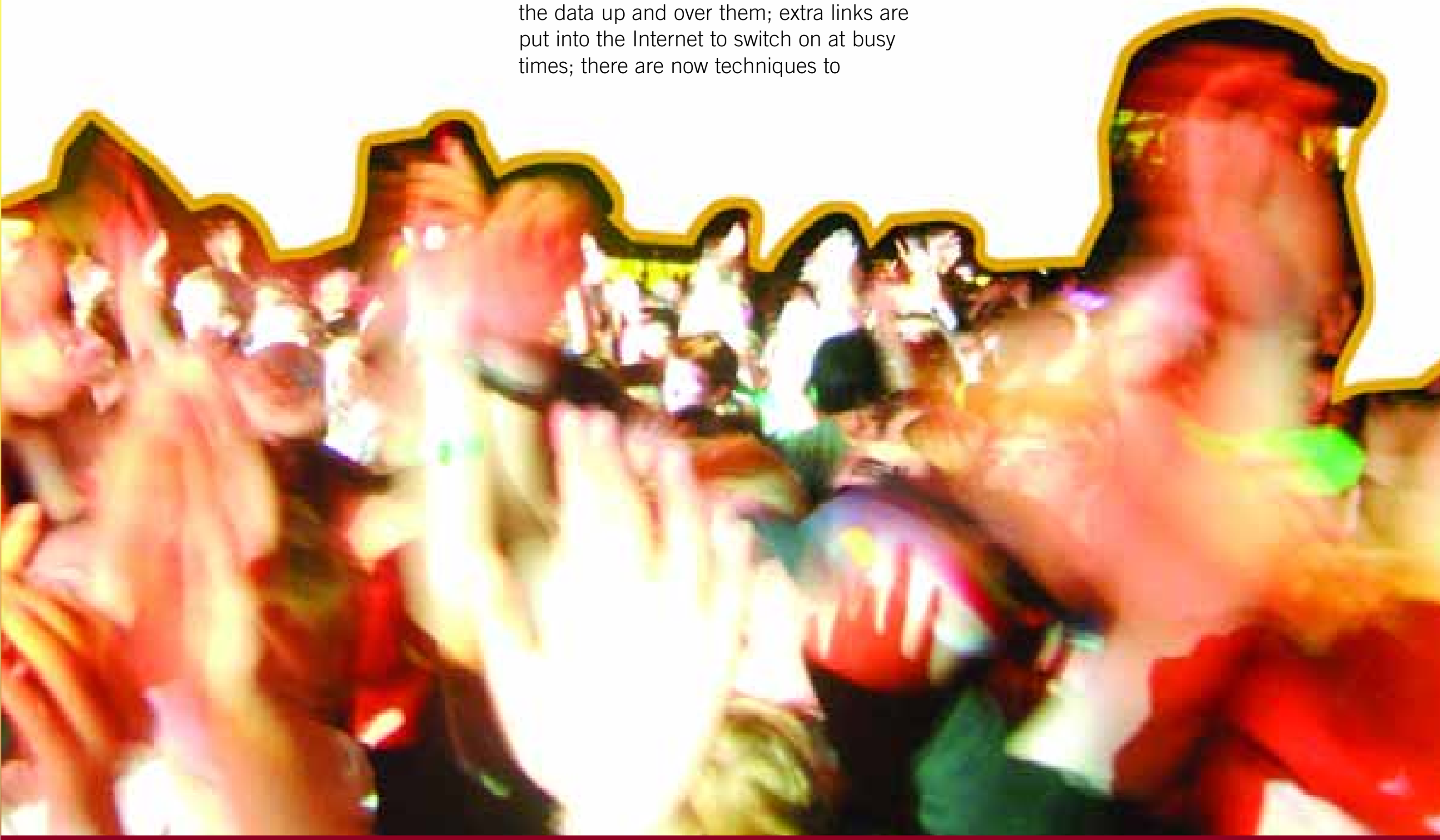
Viewers suffered as the pictures and sound cut in and out. Pictures froze then jumped. Packets arrived well after their use by date, meaning earlier images had been shown missing bits and looking fuzzy. You couldn't even recognise Madonna on stage. Some researchers found that packets had, for example, passed over seven different networks to reach a PC in a hotel just four miles away. The packets had taken the scenic route round the world, and arrived too late for the party. It wasn't only the Madonna fans who suffered. The broadcast made use of the underlying wiring of the Internet and it had filled up with millions of frantic Madonna packets. Anyone else trying to use the Internet at the time discovered that it had virtually ground to a halt and was useless. Madonna's fans had effectively crashed the Internet!

Webcasts in Vogue

Today's webcasts have moved on tremendously using the lessons learned from the early days of the Madonna Internet crash. Today video is very much a part of the Internet's day-to-day duties: the speed of the computer links of the Internet and their processing power has increased massively; more homes have broadband so the packets can get to your PC faster; satellite uplinks now allow the network to identify where the traffic jams are and route the data up and over them; extra links are put into the Internet to switch on at busy times; there are now techniques to unnoticeably compress videos down to small numbers of packets, and intelligent algorithms have been developed to reroute data effectively round blocks. We can also now combine the information flowing to the viewers with information coming back from them so allowing interactive webcasts. With the advent of digital television this service is now in our homes and not just on our PC's.

Living in a material world

It's because of thousands of scientists working on new and improved technology and software that we can now watch as the housemate's antics stream live from the Big Brother house, vote from our armchair for our favourite talent show contestant or 'press red' and listen to the director's commentary as we watch our favourite TV show. Like water and electricity the Internet is now an accepted part of our lives. However, as we come up with even more popular TV shows and concerts, strive to improve the quality of sound and pictures, more people upgrade to broadband and more and more video information floods the Internet … will the Internet Die another Day?



Robot Wars: Interview with Noel Sharkey

The television shows, Robot Wars, and its less destructive spin off, TechnoGames were extraordinarily popular and still have massive cult followings. In Robot Wars teams of contestants build their own remote controlled robots, which then attempt to shove, smash or otherwise destroy each other, ably assisted by the show's house robots, Shunt, Matilda, Dead Metal, Sergeant Bash, and Sir Killalot. In Techno Games teams built robots to compete in Olympic type events such as swimming, rope climbing, and javelin throwing – a little like a robot version of our Sodarace (see page 6). The shows have a lot in common, robots obviously, both were produced by Stephen Carsey then at Mentorn TV, and both had human judges to ensure fair play. One of the most popular judges on both was Professor Noel Sharkey of the University of Sheffield. His research background in robotics and his easy-going sense of humour made him a favourite with viewers. We caught up with a very busy Noel who agreed to do an exclusive interview for cs4fn.

What was it like working on robot wars and TechnoGames?

It was an amazing experience. The most exciting aspect was that it opened a window of communication for me with the public and allowed a dialogue about science and engineering. It also gave me a chance to really get to understand how TV works. Over 16 series of Robot Wars (including the international ones not seen here) I had the opportunity to spend time with camera operators, lighting people, sound engineers, producers, directors and a variety of presenters. Of course I enjoyed the competitions as well - especially TechnoGames. Many of the competitors were such creative engineers it was a sharp learning curve for me.

What was your most successful moment on Robot Wars?

The greatest moment that stands out for me was a kid's Robot Wars made specially for Nickelodeon in the USA. I had long been arguing with the producers about changing the immobilisation rule. That is, if even one motor stops working for at least 30 seconds, the robot is considered to be immobilized and it automatically looses. For me immobilized means that the robot has lost its mobility but I was unsuccessfully putting pressure on the production team to change the rule.

Anyway, the wheels on this kid's robot stopped turning in the middle of the competition and the house robots were sent in to finish it off. This started one of the funniest and most gratifying chases that I have ever seen. The robot had two lifting spikes at the front and it used these like crutches to hobble round. What was so good was that it managed to completely evade the house robots for twenty minutes and one of them ended up in the pit. My face said it all when the producer came to talk to me about it and the immobilisation rule was changed.

Do you think that robots have a future as TV celebrities themselves hosting game shows just like the Dr Who version of the Weakest Link?

Yes, I don't think that it would be too speculative to have a robot TV presenter. I actually worked on a kids TV programme with a robot presenter a year ago but the BBC decided not to commission it - they have really turned sour towards robots. As I said in my pitch at the time, a robot is perfect as a presenter as the main job of many presenters is to read an auto-cue while following a set route on the floor and looking into the correct camera. The robot would have no trouble remembering its lines and it will exactly follow the route that it is told to. The real issue is in giving it a personality.

You designed an emotional robot – can robots ever really have emotions?

There are a number of gestural or expressive robots around at present that can convey some of the language of human emotion. There are basically five emotions that everyone on the planet (without some mental affliction) agrees upon and can recognise: angry, sad, happy, disgusted and surprised. If you mix this with a chatty robot it will look quite convincing. But expressing emotion and feeling emotion are quite different things for a machine. I personally can't see how an inorganic object will ever feel or be aware of anything. There is a very long and technical argument behind what I am saying but I don't want to bore your readers senseless.

How did you become a Science celeb?

That is like a trick question. I don't view myself as a science celeb or any other kind of celeb (although I desperately wanted to be a famous rock guitarist in my youth). My passion is to communicate some truths about science and engineering to the general public that we should all know. I am just really lucky to have had the opportunity of a little TV fame to help me on my mission.

If someone wanted a career designing and building robots how would they go about it?

The standard answer is to take engineering, science and maths subjects at school, take an engineering degree and then get a job in the robotics industry. But my advice is to just get stuck in. There is plenty of advice on the Internet and there are a lot of little clubs and competitions around if you spend a bit of time looking.

If you like building things and want to be creative, hit the scrap yards where you can buy a lot of cheap motors for windscreen wipers, windows or seats as well as gears axels, batteries and odd bits. If you are more interested in the artificial intelligence side, buy a kit robot (or even Lego Mindstorms) and learn how to program it.

How could you make the applications of robotics of more interest to girls?

Robotics appears to be one of the bits of engineering that females are most interested in. Certainly a number of leaders in the field in the academic world are women. When I was running final year projects in computer science at Sheffield most of the few women did my projects. I wont even begin to speculate on why this might be as one of my five daughters may read it and cut my head off. But I can say that it has something to do with the multidisciplinary aspects of combining engineering, electronics and artificial intelligence. I think that there were not so many women teams in Robot Wars because of the perception of testosterone-fuelled aggression. I say "perception of" because most of the big heavy looking roboteers were real pussycats off screen.

What current research projects in robotics do you find most inspiring and why?

There are so many exciting directions in robotics at the moment that it is impossible to say which is most inspiring or most productive. We have got to a point in technological history that might later be thought of as a golden age of robotics. Robots can walk now and do all manner of acrobatics and so we just need to sort out some decent intelligence for them. For me the most inspiring thing about robotics is how they highlight how remarkable living beings are by comparison. This is one of the main reasons why I work in biologically inspired robots. There are many exciting developments and trends at the moment: humanoid robotics, producing emotional expression and developing speech and language, swarm intelligence, nano-robotics, companions, military …

Can robots be creative?

No, and another question with the same answer might be “Could a robot tell a lie?”

What got you interested in science?

To be honest, I have always been interested in everything. My family used to call it fads. One week I would be obsessed with my little microscope and the next it would be learning some instrument - I am sure that it was frustrating. My interest in science was particularly inspired by a TV programme that I used to watch in the 1950s - I can't remember what it was called - that had a nuclear physicist as one of the main characters. So that is what I wanted to be.

You have said in the past that only a biological machine is able to think. What do you think is special about the slime we are made of?

There are so many ways that I could answer this (and have done) but let me answer it with another question here. Why as a scientist should I be bothered with this "fairy tale" question? For some reason, during the foundations of AI, scientists pulled a fast one on us that wrong footed the issue ever since. In the normal course of science, the theorist puts forwards novel hypotheses that test the theory. The emphasis is on making a test strong enough to falsify the theory convincingly. When the test does not lead to falsification it provides a confirmation that will be part of an accumulation of tests that can eventually lead to acceptance of the theory (for now).

With Artificial Intelligence, we are given, instead, an in-principle argument that combines the idea that we are machines ourselves with minds that compute and the point that anything that is computable can be computed by a computer (by definition). The problem for me is that there is no evidence that our minds work like computers in the first place. The other problem is that there is not one shred of evidence that any machine anywhere has ever felt anything or seen anything (cameras record an image, they don't "see").

You will hear some scientists saying, "I know that machines can think because I can think and I am a machine" without showing any realisation of the circularity of what they are saying. It reminds me a little of the faulty logic in the syllogism, All Greeks are mortal, all humans are mortal, therefore all humans are Greek - get it? Here it is again with the right terms, All humans are machines, computers are machines, therefore computers can think. I will get off my soap-box now.

For the full interview go to the webzine.

Cash for pixels

The virtual property market on Planet Entropia is booming. Around the world thousands of fans play online in virtual worlds. In one of these multi-player role-playing games called Project Entropia gamers can buy and sell virtual items using real cash. One 23-year-old Entropia gamer spent £13,700 buying a virtual island, while another bought a virtual space station for £57,000. Real money for computer pixels, but then there is a property boom in cyber space. Other gamers who want to live on the island have paid the owner enough that he recovered his money within a year. The new owner of the space station plans to use it to start an in-game "night club" and persuade the entertainment industry to use it to sell real music and videos to gamers. Planet Entropia seems to be a great place for today's virtual entrepreneurs, so how long before the first virtual property makeover TV show appears, showing you how to increase the market price of your space station with some simple DIY computer coding and new curtains?

It started as a hobby...

ITV bought the school friends reunion website Friends Reunited for an initial £120m and will pay an extra £55m in 2009 depending upon the site's performance. Not bad for something that started as a hobby website.

Listen with Pre Vu

With perseverance and some business and technical savvy simple ideas can become award-winning inventions as Morag Hutcheon has shown.

She had the idea that when buying music in a store it should be possible to listen to the CD first. Turning her idea into reality took time though and only became possible when solid-state digital answering machine technology made the cost of producing it commercially acceptable. She also needed to develop a customised chip. That was out of the scope of her own skills, so she hired a specialist team to do it for her. The final result was Pre Vu.

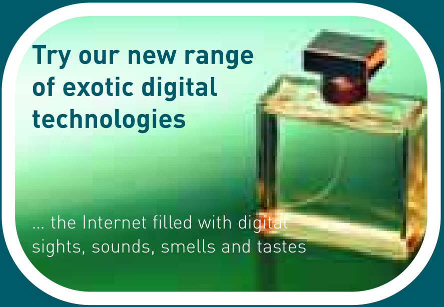
Pre Vu is integrated into the spine of a standard CD case and allows up to 60 seconds of audio content to be sampled. A personal message can also be added, for example when sending the CD as a gift, or for advertisement purposes. Morag's invention includes both the software for compressing the selected audio files and the machinery to upload the samples. As a result in February 2006, Morag was named the British Female Inventor of the Year 2006, at the British Female Inventor and Innovation Network Awards.

If you (or your mum!) are an inventive woman, you too could have your creative output nominated for these annual awards.
For more information check www.gwiin.com

Stinky computers

We all remember from art class that a whole range of colours can be made by mixing together the primary colours, red green and blue, but did you know that the same can be done with smells and tastes? Just as our eyes detect the amount of red, green and blue light being reflected from an object to give us the sensation of colours, smells and tastes can also be made by adding together primary smells and tastes: the building blocks of all we can smell and taste. Now we know what these chemical building blocks are, a whole range of exotic new digital technologies open up.

The rather strangely named iSmell system can create thousands of everyday smells from a small cartridge containing 128 primary odours. A digital signal tells the system how much of each of the chemicals to release in the same way as a computer screen produces colour by mixing red, green and blue. The smell producing chemicals need to be replaced from time to time like a printer toner cartridge to ensure that the smells produced are accurate. So using this technology you can download smells from the Internet. An intriguing thought, and one not to be sniffed at.


Try our new range of exotic digital technologies
... the Internet filled with digital sights, sounds, smells and tastes

Can we do the same with tastes? Yes. A company called TriSenx has developed a technology that allows you to 'print' tastes onto thick fibre paper sheets. Like the iSmell system it contains samples of the primary tastes that are mixed together under computer control to give the required final flavour. To enjoy the taste you simply lick the paper. Yum yum.

A whole new approach to Internet shopping might be possible with these types of technology. You can smell flowers or taste a cake before buying, or even mix your own perfumes and flavours digitally. Just think: a future on the Internet filled with digital sights, sounds, smells and tastes, not to mention the frightening possibility of spam emails that smell of spam!

Marks for the Da Vinci Code: B-

Dan Brown's book "The da Vinci Code" is one of the best selling books of all time and is now an exciting film. It's a thriller but controversially includes a claim that it is largely based on fact.

Fiction writers often change history and even go beyond the bounds of known science to get a good story, but why mangle facts for the sake of it? Especially when getting it right would have been even more interesting! Dan Brown puts silly words into the mouths of his "expert" characters: a Harvard professor, a graduate of the Royal Holloway College's leading security course, and a top historian. Even his "facts" about bee keeping are wrong.

Readers of novels should be cynical or risk believing complete fabrications - hard to do when the story is so exciting, and the overwhelming desire is to turn the page.

Even when under pressure though, a scientist should always check their facts - or they risk getting stung.

To follow the trail of the actual science go to the webzine where a real Professor, Harold Thimbleby of Swansea University, tells us why he grades it down for science.

I hear a tall dark handsome stranger...

Your horoscope for today

‘You want others to like you, but inside you tend to be critical of yourself. You can be outgoing and the *‘life and soul of the party’ but sometimes you feel reserved and simply enjoy your own company. At times you have serious doubts as to whether you have made the right decision or done the right thing.’*

Sound like you? Chances are you think it does. It’s called the Barnum Effect after the US showman P.T. Barnum and many people think it’s why horoscopes ‘work’. The Forer effect, to give it its proper scientific name, is named after the psychologist Bertram R Forer who first investigated it in 1948.

Bill Gates Horoscopes

Microsoft recently applied for a patent that involves sending horoscopes to people via their mobile phones!

Have your vote in our disappearing technology survey in the webzine. Will MP3 players still be around in 50 years?

What he found was that people thought that statements about their personality were very specific to them, when in fact the statements were vague and could apply to a whole range of people. Look at the ‘horoscope’ again, and you can see how these statements are constructed. They tend to contain generalities, and also a sort of two sidedness, ‘You seem to like the colour red, but sometimes you don’t’ sort of things.

In one TV ‘experiment’, a ‘mystic’ described the personalities of a series of women he hadn’t previously met. Afterwards they thought he had described them amazingly well. It made some believe he must really be psychic. In fact he had just given each of them the same Barnum description.

So why does it work? One of the common explanations is that the statements tend to appeal to our vanity, wishful thinking or hopes, so our brains tend to remember the parts we want to, the good bits, and ignore the parts we don’t want to hear. This sort of selectiveness in the way we store information could be related to the strategy our brains use to survive and make sense of the world. We are bombarded by information constantly, and our brains just can’t take it all in. We select what we want to store, and in the Barnum effect we filter out the parts we don’t want.

Cocktail Party

Another example of this sort of selective effect in brain processing is the ‘Cocktail Party effect’. You’ve probably experienced this yourself, the effect that is, not the party. If you’re in a crowded room with lots of people having conversations your brain ignores the voices. It becomes background noise. However the moment someone mentions your name, ‘your ears prick up’. Suddenly your brain hears something it’s interested in, ‘a good bit’, and you focus in on that conversation. In fact your brain has been processing all those voices all the time. It’s just there wasn’t anything interesting relating to you in the conversation so it was filtered out.

Just give the nod

The cocktail party effect is also being explored by researchers as a way to help people interact with mobile gadgets like mobile phones, radios and iPods without bumping into lampposts. Using 3D sound, different things can be made to appear to be coming from different positions around your head. Nodding towards the sound of lightning switches to a weather report, say, without you taking your eyes off the pavement.

By understanding how our brains select the information to look at, listen to or believe could help computer scientists develop new ways to process massive amounts of data in the future as well as being used to develop new ways of interacting with computers.

Pushy posters?

It's a long way from the man with the sandwich board standing on the street corner shouting his message about knitwear sales. London underground is planning to introduce smart posters. These interactive posters will have the ability to talk to your mobile phone. The first use for this new smart poster technology, called "Hypertags" will be to give late night travellers a phone number for safe travel information beamed direct to their mobile phones. However this technology could spread, and in the near future we could see posters for all manner of things trying to get our attention and sending us messages as we pass by.

To get a glimpse of what it might be like watch the Tom Cruise film Minority Report – the film makers employed computer scientists and other researchers to give an accurate vision of the future... well for some of it at least.

Buying Kingda Ka the eBay way

When it was first opened in May of 2005 Kingda Ka, a roller coaster in New Jersey, was the world's tallest and fastest ride. A marvel of mechanical and computer engineering it shares a weird and wonderful connection with a WWII submarine, a Volkswagen Golf car once owned by Pope Benedict XVI and Karolyne Smith's forehead. The connection is Ebay, the phenomenally successful on line auction house. The first rides on Kingda Ka and the Pope's Volkswagen Golf were auctioned on the site, and Karolyne Smith auctioned her forehead as advertisement space and was paid 10,000 dollars to have a company logo permanently tattooed there.

EBay allows the world to sell and buy almost anything as these examples show. The company was set up in 1995 by Pierre Omidyar in his back room, so the story goes, and is among the fastest-growing companies of all time. Buyers and sellers communicate through the web site, and eBay makes a profit by charging them to do so.

With millions of users worldwide, and no personal contact, trust is an important factor. EBay use a form of self-policing where other users can rate buyers on their trustworthiness. This method is needed to keep things as safe as possible but also to allow the service to expand. It could never have expanded so rapidly if eBay had to employ staff to check each sale. Though this self-policing has sometimes been controversial, and eBay has had its share of scams, it is still extremely popular. Many users admit to being hooked on rummaging through the pages and looking for bargains, and there are certainly some wonderful bargains to be had.

But finally, if you're trying to sell a decommissioned aircraft carrier, you should know that the last one put up for auction on ebay Motors didn't sell. Don't say you haven't been warned.

Fancy some Green chips?

Today, energy efficiency is a vital part of any new technology – to protect the planet everything we do must be sustainable. For example, computer scientists and engineers at Intel are working to try and make your chips cooler. The silicon chip in your laptop or mobile phone needs electricity. The chip is made up of thousands of microscopic electronic switches that allow the computer program to do the calculations to make your application work. Normally all these switches click over at the same time. It's called a clock cycle. Your data goes in. Click. The data is processed. The data comes out. The problem is that these chips, particularly if they run quickly by having a high-speed clock, get very, very hot. This heat is due to the current in the circuits that causes the materials to heat up. All this heat shortens your battery life and is very inefficient.

Intel's cunning plan is to build a chip that adapts. It's called SpeedStep technology. When not much is going on, the clock runs slowly conserving your battery. Only when lots of data needs processing does the chip turn the clock speed up. This clever chip is just one way that today's computer scientists are trying to build eco-friendly computers, and produce chips that wont fry your battery.

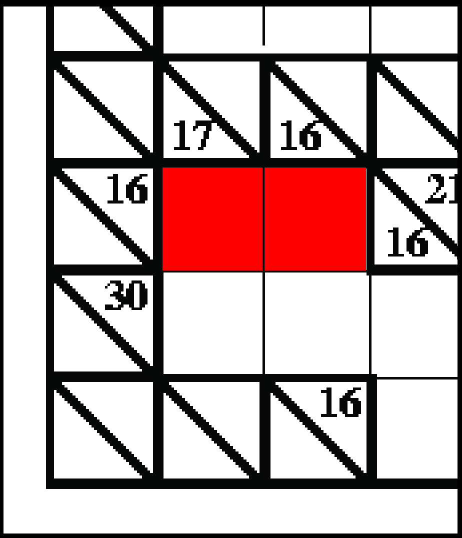
Kakuro, Sudoku and Computer Science

To be a good computer scientist you have to enjoy problem solving. That is what it's all about: working out the best way to do things. You also have to be able to think in a logical way: be a bit of a Vulcan. But what does that mean? It just means being able to think precisely, extracting all the knowledge possible from a situation just by pure reasoning. It's about being able to say what is definitely the case given what is already known...and it's fun to do. That's why there is a Sudoku craze going on as I write. Sudoku are just pure logical thinking puzzles (most of which are generated by computers of course). Personally I like Kakuro better: similar to Sudoku, but with a crossword format.

What is a Kakuro?

A Kakuro is a crossword-like grid, but where each square has to be filled in with a digit from 1 to 9 rather than a letter. Each horizontal or vertical block of digits must add up to the number given to the left or above, respectively. All the digits in each such block must be different. That part is similar to Sudoku, though unlike Sudoku, numbers can be repeated on a line as long as they are in different blocks. Also, unlike Sudoku, you aren't given any starting numbers, just a blank grid.

Where does logic come into it? Take the following fragment:



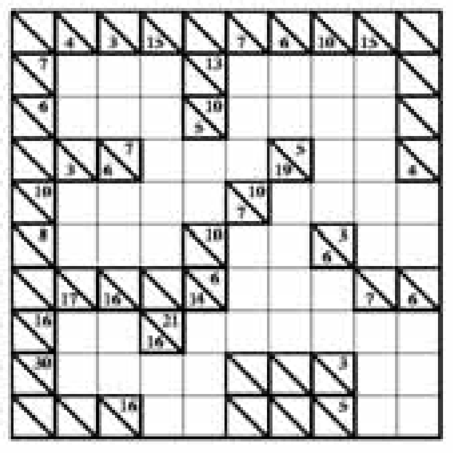
There is a horizontal block of two cells that must add up to 16. Ways that could be done using digits 1 to 9 are 9+7, 8+8 or 7+9. But it can't be 8+8 as that needs two 8s in a block which is not allowed so we are left with just two possibilities: 9+7 or 7+9. Now look at the vertical blocks. One of them consists of two cells that add up to 17. That can only be 9+8 or 8+9. That doesn't seem to have got us very far as we still don't know any numbers for sure. But now think about the top left hand corner. We know from across that it is definitely 9 or 7 and from down that it is definitely 9 or 8. That means it must be 9 as that is the only way to satisfy both restrictions.

Here is a full Kakuro to try.

Check you got it right on the cs4fn website when you are done.

Being able to think logically is important because computer programming is about coming up with precise solutions that even a dumb computer can follow. To do that you have to make sure all the possibilities have been covered. Reasoning very much like in a Kakuro is needed to convince yourself and others that a program does do what it is supposed to. An ongoing challenge is in developing programs that can do that kind of reasoning and so be able to tell us whether other programs are correct or not.

For more logic puzzles go to the cs4fn webzine.



Taking a RISC from the logic piano

William Stanley Jevons was born in Liverpool in 1835. He was famous in his day as an economist and his smash hit book 'The Coal Question' called the nation's attention to the reduction in Britain's coal supplies. Jevons had other strings to his bow though and one of the strangest was his "logic piano". Jevrons was fascinated with logic and reasoning. He believed you could start with one thing (a premis) and from this work through a chain of reasoning to the conclusion, and that this could be done for everything. So he set about building his wooden Logic Piano, where you could put in the premises, play the keys to mechanically apply his reasoning rules, and discover the conclusion. Amazingly it did work and is similar in idea to modern day theorem provers used to verify properties of computer designs. Of course, being small and woody, it couldn't solve every thing but then it turns out that was always an impossible dream (see page 58).

RISC chips are everywhere; in Play-stations, iPods, mobile phones...

As the years passed others took on his idea, thankfully turning from wood as the material of choice, till finally the silicon microchip was developed. No longer timbered, it could take the electronic signals (we can think of these as the premises) and output the conclusion by following lots of instructions on the chip, like the piano did. As the chips got bigger it turned out that lots of the memory space on the chips wasn't being used well. Chips were designed for storing big numbers but most applications used smaller numbers, so the space for storing these on the chip, the registers, were often not completely filled, and too much time was being spent on importing data from outside memory onto the big empty spaces on the chip. Worse still the instructions just became too big and slow to carry out sensibly. Economic madness as Jevons would no doubt have said!

Enter RISC (Reduced Instruction Set Computers): economical and not a piano key in sight. It uses the memory on the chips economically, and rather than use complicated instructions it instead uses many simple instructions. Musically it's like moving away from trying to orchestrate a concerto when it was easier to simply pluck the notes. Today RISC chips are everywhere; in play-stations, iPods, mobile phones…You name it it's probably got a RISC chip in it computing efficiently away. RISC chips are one of the key reasons behind the spread of computers away from offices and into our everyday lives. The fine-tuning still isn't over yet though. Multinational companies like ARM still work to find the best way to play the most efficient logical tunes on their RISC machines. It's a good guess that this constructive use of time and resources would be music to Jevons' ears.

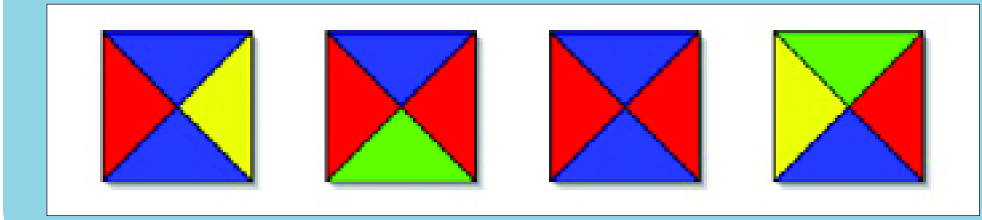
The Uncomputable Jigsaw Puzzle

Your missions should you choose to accept them...

1) First a simple jigsaw puzzle…Can you work out how to put the following square tiles into a 5x3 rectangle so that touching sides of tiles match. You can use as many of each tile as you like but no different ones.

2) Now a little harder. Can you say whether or not it's possible to tile a rectangle of any size (with dimensions a multiple of tiles of course) with those 4 tile patterns using any number of each? Touching sides must always match.

3) Finally, the tough bit, can you write a set of instructions that give a way, if followed blindly, of solving the above problem for any tile set…Whatever tile patterns you start with, your instructions must say whether they can tile any sized floor or not.



Have we set you a Mission Impossible? Read page 58 to find out.

Mission:Impossible

In each episode of Mission:Impossible, the M:I team have a seemingly impossible task to perform, which of course they duly do. They achieve these feats using not only Tom Cruise/Ethan Hunt heroics and clever plans but also hi-tech gadgets that help make the impossible mundane. Is there nothing that hi-tech coupled with smart people can't overcome once we set our minds to it?

The massive progress being made as computer science rapidly changes the way we live makes it seem anything is possible. Computers can now fly a plane across the Atlantic, including taking off and landing, with no human intervention. They can beat the best human at chess. They can store my whole music collection in my pocket. Computers can recognize faces, tell me which way to drive to get home ... When in the past people have made predictions about future technology things that would never happen they have tended to end up looking foolish. With a bit of ingenuity in the future computers will solve any problem we want them to ... won't they?

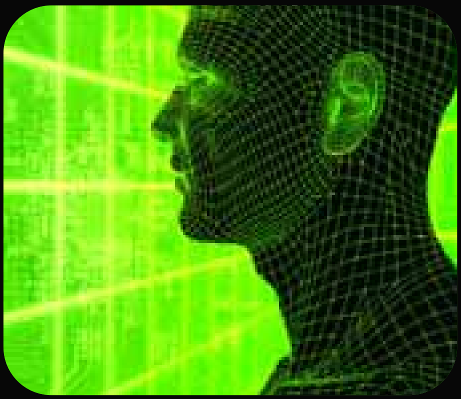
Lots of problems, are 'uncomputable' - they can never be solved even by hyper-intelligent beings from the planet Vorg

It turns out though that some missions really are impossible for computers and even Tom Cruise wouldn't be able to make a difference even if he was given unlimited time. Not now, not ever...and just to hammer it home, this is a fact that was proven mathematically way back in the 1930s before any one had even created an actual working computer...

Computer Science is not just about what computers can do, but also about what they can never do, which is where the bad news for the Mission:Impossible team comes in. Computer scientists have shown that lots of problems that must have solutions are 'uncomputable' - they can never be solved using computers however powerful. Uncomputable problems are not just ones that computers can't solve either, humans can never know the answers, nor can hyper-intelligent beings from the planet Vorg.

What kinds of problems? You might hope that they would be ones that sound impossible from the outset, like cracking the Dr Who Skasis Paradigm that will give you power over the whole of time perhaps? It turns out that even some innocuous problems will never be solved (see page 57 for one to do with Jigsaws)...and in some cases they go to the heart of what we would like computers to be able to do for us like guaranteeing the programs we write do what they are supposed to, or proving mathematical facts from premises (see page 57).

...impossible even if he was given unlimited time...



So remember, if you ever join the Mission:Impossible Team, there are some missions that, once that tape has self-destructed, the only thing to do is just say "cop that for a lark" and go home. The good news is there are still lots of exciting computer science missions that aren't impossible...should you choose to accept them.

It's a secret Deal or No Deal

A hit TV gameshow with 22 sealed boxes and just one question: Deal or No Deal? Enough money to buy a mansion? Or just a jelly baby? How is your nerve? Does your friend Ross really know what's in his box? Will you take the money on offer from the Banker, or wait to see what finally is in your box? With the limited information you have of what prizes have gone and advice from the other colourful players, do you make the deal or not?

The rules of TV show 'Deal or No Deal' are simple; the game is addictive viewing. It is the hidden information of the boxes combined with the other players' personalities that make it compelling. In fact the show's producers put all that week's players in the same hotel so they can get to know each other before the game. They get to know each other's personalities.

Computer Science or No Computer Science

So what does it have to do with computer science? The most obvious link is that the game play is just like a computer program, it follows rules…but another link is in the hidden information and the way that participants can makes guesses on the contents of the boxes depending on the players' personalities. It's like the boxes and the personalities are the same thing at times.

Noel or No Noel?

Noel Edmonds runs the show, but he isn't allowed to do anything. He just has to make sure the game follows the format: follows the rules. To create an actual run of the show you need more than just the rules of course. You need props and personalities… resources to manipulate…boxes and contestants. In computer programs the equivalent are variables. Variables have names to refer to in place of the hidden values within, just like the boxes have numbers 1 – 22 so you can refer to a given box, its cash value inside unknown. Variables store the information that the program manipulates as it runs, just as Noel, following the rules, manipulates the contestants and the boxes. The rules say boxes can only be opened at specific points and with conditions attached about what happens to the money within. Just like repeatedly running a computer program, every show plays out differently even though the steps followed are always the same.

Crash or No Crash

Following the intended rules is critical. If a box is accidentally opened at the wrong time everything goes haywire. That happened in April 2006 when a contestant called Tom dropped his box, revealing the contents when they should have been hidden. The whole show had to be interrupted to sort out the mess. Dropping the box was like a maverick rule no one had noticed suddenly being followed – a bit like a previously unknown bug in a running program crashing it. It would have to be fixed on the fly with new code added and the program restarted. In Tom's case the new "code" added was a rule that if a box is dropped all are reshuffled with their values redistributed. Keeping the information hidden was everything.

> **Dropping the box was like a maverick rule no one had noticed suddenly being followed**

Group or No Group

There is more to the boxes than their hidden money values though, and that is where a different kind of information hiding comes in: "encapsulation"…grouping things and temporarily hiding the details of what has been grouped. Deal or No Deal boxes are more than just a number they have personalities too. Each box is linked to a contestant. Even though the people and the boxes are completely different things, you don't really think of them like that but as a single entwined thing. Why pick a particular box to be opened? Because its contestant has been lucky in the past. If Ryan's box has a low value in, then you remember that about Ryan in future shows. When you think of Ryan you mean his box too even though you don't spell it out.

Toby or Not Toby

This idea of grouping different things in our minds as single animate objects is a very natural thing we do and it's helpful to programmers too. This idea of encapsulation is one of the key things about the style of programming know as "Object-oriented programming". That is all an "object' is to a programmer. A bunch of related resources with their own properties and abilities, glued together, to be thought of as one thing. Why is it useful? Because it is easier to keep track of just one thing than lots of related ones, and it's a natural way for us to think about both the world and the programs we write. Talk about Toby and, without spelling it out, you mean his personality, his history over the previous weeks, his current box and the value in it. When Toby moves, his box moves with him. You only think about the separate parts when you need them. When it's time to open Toby's box, then you can focus on the box alone rather than the personality. In a game show it makes the game captivating. In programming it makes the program easier to write. With millions of lines of rules to write that's serious.

Back (page) in fashion

Clothes and accessories make up the fashion conscious world we live in today. Computing devices already form part of the wardrobe of many. Being seen with the right mobile phone, or the stylish white iPod headphones makes a statement about you. But what does the future hold for fashion? We take a stylish peak into the wardrobe of the future at the clothes and accessories to come.

Wearable Computers: a fashion statement?

You can actually wear a computer today if you want to. You can buy a pair of eyeglasses that look like ski goggles and act as a monitor and a hard disk drive. They have a wireless connection to the eyeglasses that you can strap on your wrist. It's been suggested that using these sorts of technologies can record everything we see. Sort of like your own personal aeroplane black box, so you'll never forget those fascinating memories of a rainy day at the shops. Whether we would want to do this is another matter! We also might not want to wander around with 'bits of computers' visibly hanging on us. Of course for some it will be a cool look but for others it would be a 'bolt on' fashion no-no.

Catwalk comment: a little too much bulky Borg this year.

A new meaning for smart clothes

We are used to saying that someone is a smart dresser, but when technology comes into fashion the words 'smart clothes' take on a whole new meaning. Smart in the technology sense means intelligent, and new fabrics and clothing are being developed which have the ability to sense and change depending on the world around them.

Catwalk comment: clever calculating clothes can create crazy combinations.

Smart to be Safe

An example of where this smart technology can go are a new generation of hazard suits that are equipped with sensors that keep an eye on your position and measure your vital signals, such as temperature and heart beat. Applications would be for teams of people who brave disaster areas. The information from the suits can be sent via a wireless link to a central control where the health of the rescuers can be monitored. The information can also be passed to other people in the team and combined with useful bits of data such as maps or weather reports, to ensure the team have the best chance to work together effectively, efficiently and safely.

Catwalk comment: clothes that know where you are going take a fashion lead.

Electronics that wear well

Carrying your mobile phone, laptop or PDA's may become a thing of the past when technology goes into your clothes. A Germany company already sell a jogging outfit, which has an MP3 player in the sleeve, activated by your voice command through a microphone in the collar. There are also new washable fabrics that have electronic circuits woven into them so in effect the design on your coat could become a keyboard, and all the other electronic parts could miniaturise and vanish into pockets, in effect you are wearing your phone, iPod or computer… perhaps even all of them at the same time.

Catwalk comment: this collection may be music to your ears.

A phone with your finger-tips

In the future when you are being phoned, you may simply feel a tickle in the phone band on your arm. Click your fingers then stuff your finger in your ear to talk to friends! Strange as it seems a Japanese company are developing a system that sends vibrations from your phone armband down your arm and through your fingers to provide the sound to act as an earpiece.

Catwalk comment: using your finger to phone gives a whole new meaning to digital phones.

Sick of clothes?

There are plans to develop biometric bodysuits that will measure your vital signs and where appropriate dispense medicine to you; it will be like wearing a Doctor 24/7. As with all new technology we need to decide if this is something we would be happy with. Is it right that a smart suit decides your prescription, or would you feel happier to have human doctors making these important health decisions?

Catwalk comment: the smart way to get better.

Weaving our future with computers

Over the next decades billions of interacting microscopic computers will vanish into the background, becoming part of the weave of modern life. Clothes will get smarter, cool new technologies will change the way we live our lives and designers, artists and computer scientists will work together to fashion a brighter future for us all.

See the webzine's Magazine+ www.cs4fn.org for extended versions of this and other articles

cs4fn is produced by Paul Curzon and Peter McOwan of the Department of Computer Science, Queen Mary, University of London with help from bunches of different people. This annual was made possible with a grant from EPRSC.

EPSRC Engineering and Physical Sciences Research Council

BCS

intel

Microsoft

ARM

Queen Mary
University of London