ISSN 1754-3657 (Print)
ISSN 1754-3665 (Online)

CS4FN

Computer Science for Fun

Issue 18

Machines That Are Creative

The Sorceror's Apprentice 2.0

Can a computer tell a good story?

Music-making mates

Queen Mary
University of London

Can machines be creative?

In this issue we explore whether machines can be creative. Ada Lovelace suggested in the 1800s that one day they might, and now computational creativity researchers are making it happen. We look at the first attempt at a creative algorithm for writing love letters and more recent programs that generate novel stories and funny tweets. Machines also create music: from programs that evolve better music to ones using it to improve their relationships with humans. We even look at programs that intend to paint portraits and artificial intelligences trying to create magic tricks. Whatever kind of art we may want to create, the computers are having a go at creating it too.

Ada Lovelace: Visionary

by Paul Curzon, Queen Mary University of London

It is 1843, Queen Victoria is on the British throne. The industrial revolution has transformed the country. Steam, cogs and iron rule. The first computers won't be successfully built for a hundred years. Through the noise and grime one woman sees the future. A digital future that is only just being realised.

Ada Lovelace is often said to be the first programmer. She wrote programs for a designed, but yet to be built, computer called the Analytical Engine. She was something much more important than a programmer, though. She was the first truly visionary person to see the real potential of computers. She saw they would one day be creative.

Charles Babbage had come up with the idea of the Analytical Engine - how to make a machine that could do calculations so we wouldn't need to do it by hand. It would be another century before his ideas could be realised and the first computer was actually built. As he tried to get the money and build the computer, he needed someone to help write the programs to control it - the instructions that would tell it how to do calculations. That's where Ada came in. They worked together to try and realise their joint dream, jointly working out how to program.

Ada was a mathematician with a creative flair and while Charles had come up with the innovative idea of the Analytical Engine itself, he didn't see beyond his original idea of the computer as a calculator. Ada saw that they could do much more than that.

The key innovation behind her idea was that the numbers could stand for more than just quantities in calculations. They could represent anything - music for example. Today when we talk of things being digital - digital music, digital cameras, digital television, all we really mean is that a song, a picture, a film can all be stored as long strings of numbers. All we need is to agree a code of what the numbers mean - a note, a colour, a line. Once that is decided we can write computer programs to manipulate them, to store them, to transmit them over networks. Out of that idea comes the whole of our digital world.

Ada saw even further though. She combined maths with creativity and so she realised that not only could they store and play music they could also potentially create it - they could be composers. She foresaw the whole idea of machines being creative. She wasn't just the first programmer, she was the first truly creative programmer.

Ada also wrote

"The Analytical Engine has no pretensions to **originate** anything."

So how does that fit with her belief that computers could be creative? Read on and see if you can unscramble the paradox.

Playing with robots

by Paul Curzon, Queen Mary University of London

There are many, many toys to play with these days that contain computers. That means toys are getting more intelligent. Toys are starting to be able to play with us rather than just being played with. But why do children play? Given how much we do it, play must be important! Many animals play too and often seem to be having as much fun as us. To have evolved, play must increase their chances of survival somehow. Could it be giving a creative edge?

Play is partly about practising skills that will be useful in the future, partly about building strength. A kitten that constantly plays at pouncing on things will become a much better hunter. An active child will grow up fitter than a couch potato who never runs around a playground. Is there more to it than that though? Patrick Bateson of the University of Cambridge and Paul Martin of King's College London think it could be about creativity too.

We mean lots of different things when we talk about play – playing in your first Wimbledon final is all-together different from having a fun pillow fight. Similarly, when Gareth Bale is bearing down on goal having stormed the length of the pitch, it's unlikely he's feeling playful. That kind of aggressive, competitive play doesn't seem to be linked to creativity. Playful play is though. What matters for creativity seems to be a playful mood – feeling positive and light-hearted – more than the activity itself.

Researchers have found that the more playful kinds of animal, like dolphins, that seem to play for the sake of playing, are also the most creative in coming up with novel ways to do things. Dolphins naturally play in dozens of ways: from playing with bubble rings to playing with seaweed. They are also great at coming up with new things to do and new ways to solve problems set them.

What matters for creativity is a playful mood.

By playing, humans and animals alike may have more chance to experience situations and generate ideas that will help to solve novel problems in the future. They may be exercising their brain as well as their muscles. In the wild, for example, dolphins often use curtains of bubbles to trap fish to eat – just like the ones they play with! Perhaps it was bubble play that gave them the fishing idea.

Play does seem to enhance the creativity of children. The jury is still out on whether it leads to adults being more creative too – lots more research is needed to find that out for sure. What is certain is that successful companies don't just want to employ hard-working intelligent people. They want their workers to be creative too and many high tech companies that rely on innovation, and so the creativity of their staff, already actively try and foster playfulness. Google is perhaps the best example. The company plays with its logo daily and its offices are famous for the playful atmosphere. You do not have to stop playing just because you grow up. People who see work as a place to play may well be the most valuable employees a company has.

Now, if play is important in helping both humans and animals be creative, then perhaps we should be creating machines that are playful too. Perhaps the intelligent machines of the future will need to play to achieve their own potential just as much as we do. Robots are likely to be our children's playmates in the future. Maybe the robots will need playmates as part of their growing up too.

The algorithm that could not speak its name

by Paul Curzon, Queen Mary University of London

The first program that was actually creative was probably written by Christopher Strachey, a descendent of first programmer, Ada Lovelace, in 1952. It wrote love letters…possibly gay ones.

The letters themselves weren't particularly special. They wouldn't make your heart skip a beat if they were written to you, though they are kind of quaint. They actually have the feel of someone learning English doing their best but struggling with the right words! It's the way the algorithm works that was special. It would be simple to write a program that 'wrote' love letters thought up and then pre-programmed by the programmer. Strachey's program could do much more than that though – it could write letters he never envisaged. It did this using a few simple rules that despite their simplicity gave it the power to write a vast number of different letters. It was based on lists of different kinds of words chosen to be suitable for love letters. There was a list of nouns (like 'affection', 'ardour', …), a list of adjectives (like 'tender', 'loving', …), and so on.

It then just chose words from the appropriate list at random and plugged them into place in template sentences, a bit like slotting pieces into a jigsaw. It only used a few kinds of sentences as its basic rules such as: "You are my <adjective> <noun>". That rule could generate, for example, "You are my tender affection." or "You are my loving heart", substituting in different combinations of its adjectives and nouns. It then combined several similar rules about different kinds of sentences to give a different love letter every time.

Strachey knew Alan Turing, who was a key figure in the creation of the first computers, and they may have worked on the ideas behind the program together. As both were gay it is entirely possible that the program was actually written to generate gay love letters. Oddly, the one word the program never uses is the word 'love' – a sentiment that at the time gay people just could not openly express. It was a love letter algorithm that just could not speak its name!

Diamond Dogs

Rock star David Bowie co-wrote a program that generated lyric ideas. It gave him inspiration for some of his most famous songs. It generated sentences at random based on something called the 'cut-up' technique: an algorithm for writing lyrics that he was already doing by hand. You take sentences from completely different places, cut them into bits and combine them in new ways. The randomness in the algorithm creates strange combinations of ideas and he would use ones that caught his attention, sometimes building whole songs around the ideas they expressed. An algorithm is the reason his song lyrics are so surreal!

Rules of Engagement

by Tony Veale, University College Dublin

When you see a child throwing a tantrum on a train, who do you blame: the child, who – though annoying – may not know any better, or the parent, inured to the noise and unwilling to do anything about it? Thinking is something we must all learn to do if we are to do it well, and we must all learn to think socially as well as intellectually to successfully engage with the world. The same is true for our machines: to engage successfully with the world, they must engage successfully with us and with each other. They are like children: when they produce hilariously stupid results, the fault lies as much with us as with them. If we don't train them to engage with the task at hand, to anticipate the unexpected, and to know when their rules are about to break down, then we all share in the stupidity that ensues.

Bake me a cake as fast as you can

Even bakers now use technology. Email them the picture or text you want on your cake, and they use a special printer (with food dyes for ink) to print your design onto the icing.

This real cake design produced by a New York bakery is a funny example of what can go wrong when people and machines don't engage properly with what they are doing.

The customer used Microsoft Outlook to email the cake message to the bakers. It inserts special HTML tags (code used to format web pages) into its emails to make them look prettier. Unfortunately, the bakery doesn't use Outlook, so the HTML was cut-and-pasted directly into the cake-printing program. We can laugh at the software but the human baker is mostly at fault. Did they really think this was the cake design? Poor Aunt Elsa. If a person can't detect this kind of problem, what hope for our machines?

Breaking Rules

It is commonly believed that creativity is about breaking rules. If you are going to break a rule, start with that one! Nothing is further from the truth. Creativity comes from a hyper-understanding of the rules rather than from a willful ignorance of them. We must know the limits of our rules, and how to tell the difference between a convention and a hard rule. People are creative all the time in chess, poetry and soccer without ever breaking the rules: instead, they break with convention! Creativity requires a deep engagement with the rules, to know where individual initiative can take over. Can computers ever show this kind of engagement with rules? The answer is a qualified "yes": they must be programmed in the right way, not just with hard-coded rules, but with knowledge of their own workings, able to reason about their own rules. Building systems like that is what computational creativity is all about. Ironically, such programs may have more self-knowledge than a human doing the same job. By using introspection to design them we humans can obtain greater self-knowledge of how we ourselves work.

To find out more about computational creativity research check out RobotComix.com which is full of computational creativity comics and cartoons, including a new book Hand-Made By Machines. Enjoy!

Yeah, But is It?

by Simon Colton, Goldsmiths College

“Yeah, But is it Art?” How many times do we hear this? In magazines, web forums, Twitter, TV and radio: from Damien Hirst’s pickled sharks and Tracey Emin’s unmade bed, to whether videogames are art. The argument is never ending and often bitter. Thank goodness for that! Imagine how much more boring life would be if we all suddenly agreed. As a society, we have agreed to disagree about art, and the resulting arguments drive innovation enhancing our world.

Back in the 1950s, the philosopher Gallie introduced the idea of essentially contested concepts: concepts that lead to endless arguments that can’t be settled by logic or collecting evidence, ones where everyone claims the others are using the concept the wrong way. There are things in life that we are meant to disagree about! How interesting is that? And scary, especially for scientists trying to study such a disputed concept.

It’s no wonder that nobody has come close to defining what creativity is, or to explaining why some people are more creative than others: it is an essentially contested concept. We need to disagree about creativity so it can be a driving force for change. But it raises the serious question of how to get software to act in creative ways, and to get society to accept machines are creative, if we’re not prepared to define what creativity is. Most other areas of Artificial Intelligence are based on concrete definitions, with increasingly sophisticated software written to perform intelligent mini-miracles safely within those definitions. But, in Computational Creativity research, if we embrace discord and uncertainty about the key concept, how can we proceed scientifically?

All is not lost. Some of us have been concentrating on the perceptions that people have of software being creative or not, with specific emphasis on the ‘or not’ part. It’s easy to dismiss software as not being creative, and people’s reasons tell us how we can change their views, enabling them to appreciate more what software does, and what wonderful things it produces. There is no creativity gene or algorithm, only the perceptions of people. Think of a white wall. In the day, it’s obviously a white wall, isn’t it? Go back at night with the lights off, and that white wall will be as black as can be. The change in circumstances has changed the perception you have of it. The same is true of human qualities, like being funny or being creative. And the same is true of computer programs. So, if we can make it hard for people to perceive software as not creative, then maybe, just maybe, they might one day be prepared to call it ‘creative’.

One argument is that software is uncreative, because the programmer supplies all the intentionality. “You write and run the software”, they say, “it’s you that wants to create something – so it’s not being creative: you are!” And they are right, but that can change. By taking these criticisms, we can write software that addresses the issues and makes it difficult to claim that the program is not creative (for example, see the Painting Fool, on the opposite page, and intentionality). “It’s still not creative”, they say, “because it’s…erm… I mean, it’s not… erm…”. That’s the hope, anyway. **Only time will tell.**

The Painting Fool

by Simon Colton, Goldsmiths College

Can a machine intend to create a work of art? We looked at this during a Paris art exhibition called 'You Can't Know my Mind' with a program called 'The Painting Fool' (**www.thepaintingfool.com**). We want it to be taken seriously as a creative artist in its own right, one day. At the exhibition, we explored whether people would think of the software as intending to do things.

The Painting Fool painted portraits...but only if it felt like it. When someone sat to have their portrait painted, it was reading the newspaper. If you were unlucky enough to ask for a portrait when it had just read lots of truly miserable articles, it told you to go away, explaining that it didn't think it was appropriate to paint a portrait when it was in such a (simulated) bad mood. It would send you packing with a miserable quote from a miserable article, and suggest you come back later, when it might be in a better mood. In the context of 'You Can't Know my Mind', we hoped that this would emphasise its independence. In fact, several times, we desperately wanted it to paint a portrait for some VIP, but it refused and we had to live with that decision.

For people who caught the software in a not-so-terrible mood, The Painting Fool would attempt to paint a portrait which captured its mood. If in a good mood – after reading lots of upbeat articles – it asked the sitter to smile. It made clear that it was using them as a model, rather than them using it as a drawing tool. It took a photo for the portrait then made a quick sketch of what it wanted to achieve. It then set about simulating the use of pencils, paints and pastels. It made various choices while painting that it hoped would enhance the chances of people using a particular word like "bright", "colourful" or "crazy" (if in a good mood) or "dull", "bleary" or "grey" (if in a bad mood) to describe the portrait.

At the end, The Painting Fool took a long hard look at the portrait, to check whether its picture matched the mood it hoped to portray. Finally it told the sitter how well it thought it had done, and whether this was a "great success", "miserable failure" or somewhere in-between. It also learned from its successes and its mistakes, so that it became more likely to achieve the required mood with future portraits. We asked people whether they thought the software had exhibited a little intentionality. Now we ask you: what do you think?

And yeah, but. Is it Creativity?

Can a computer tell a good story?

by Rafael Pérez y Pérez of the Universidad Autónoma Metropolitana, México

What's your favourite story? Perhaps it's from a brilliant book you've read: a classic like Pride and Prejudice or maybe Twilight, His Dark Materials or a Percy Jackson story? Maybe it's a creepy tale you heard round a campfire, or a favourite bedtime story from when you were a toddler? Could your favourite story have actually been written by a machine?

Stories are important to people everywhere, whatever the culture. They aren't just for entertainment though. For millennia, people have used storytelling to pass on their ancestral wisdom. Religions use stories to explain things like how God created the world. Aesop used fables to teach moral lessons. Tales can even be used to teach computing! I even wrote a short story about a kidnapped princess to help my students understand things like bits.

It's clear that stories are important for humans. That's why scientists like me are studying how we create them. I use computers to help. Why? Because they give a way to model human experiences as programs and that includes storytelling. You can't open up a human's brain as they create a story to see how it's done. You can analyse in detail what happens inside a computer while it is generating one, though. This kind of 'computational modelling' gives a way to explore what is and isn't going on when humans do it.

So, how to create a program that writes a story? A first step is to look at theories of how humans do it. I started with a book by Open University Professor Mike Sharples. He suggests it's a continuous cycle between engagement and reflection. During engagement a storyteller links sequences of actions without thinking too much (a bit like daydreaming). During reflection they check what they have written so far, and if needed modify it. In doing so they create rules that limit what they can do during future rounds of engagement. According to him, stories emerge from a constant interplay between engagement and reflection.

What knowledge would you need to write a story about the last football World Cup?

With this in mind I wrote a program called MEXICA that generates stories about the ancient inhabitants of México City (they are often wrongly called the Aztecs – their real name is the Mexicas). MEXICA simulates these engagement-reflection cycles. However, to write a program like this you need to solve lots of problems. For instance, what type of knowledge does the program need to create a story? It's more complicated than you might think. What knowledge would you need to write a story about the last football World Cup? You would need facts about Brazilian culture, the teams that played, the game's rules… Similarly, to write a story about the Mexicas you need to know about the ancient cultures of México, their religion, their traditions, and so on. Figuring out the amount and type of knowledge that a system needs to generate a story is a key problem a computer scientist trying to develop a computerised storyteller needs to solve. Whatever the story you need to know something about human emotions. MEXICA uses its knowledge of them to keep track of the emotional links between the characters using them to decide sensible actions that then might follow.

By now you are probably wondering what MEXICA's stories look like. Here's an example:

Jaguar Knight made fun of and laughed at Trader. This situation made Trader really angry! Trader thoroughly observed Jaguar Knight. Then, Trader took a dagger, jumped towards Jaguar Knight and attacked Jaguar Knight. Jaguar Knight's state of mind was very volatile and without thinking about it Jaguar Knight charged against Trader. In a fast movement, Trader wounded Jaguar Knight. An intense haemorrhage aroused which weakened Jaguar Knight. Trader knew that Jaguar Knight could die and that Trader had to do something about it. Trader went in search of some medical plants and cured Jaguar Knight. As a result, Jaguar Knight was very grateful towards Trader. Jaguar Knight was emotionally tied to Trader but Jaguar Knight could not accept Trader's behaviour. What could Jaguar Knight do? Trader thought that Trader overreacted; so, Trader got angry with Trader. In this way, Trader - after consulting a Shaman - decided to exile Trader.

As you can see it isn't able to write stories as well as a human yet! The way it phrases things is a bit odd, like "Trader got angry with Trader" rather than "Trader got angry with himself". It's missing another area of knowledge: how to write English naturally! Even so, the narratives it produces are interesting and tell us something about what does and doesn't make a good story. And that's the point. Programs like MEXICA help us better understand the processes and knowledge needed to generate novel, interesting tales. If one day we create a program that can write stories as well as the best writers we will know we really do understand how humans do it. Your own favourite story might not be written by a machine, but in the future, you might find your grandchildren's favourite ones were!

For Rafael's short story 'A Godlike Heart' go to **www.cs4fn.org**. If you like to write stories, then why not learn to program too then you could try writing a storytelling program yourself. Could you improve on MEXICA?

The judge of a good story

A storyteller must be able to judge their work. Rafael created a system that could evaluate stories. It's not easy. What makes a good story? It's questions like this he's trying to answer. His program evaluates the interestingness, coherence and novelty of a story. The following is its evaluation of the Jaguar Knight story: do you agree with it? If not can you explain why not?

This story is really good! I like it! Congratulations!

Here are some comments about your work that I hope will be a useful feedback.

Coherence

The story is very logical; all actions are nicely integrated and form a coherent unit.

At the end most difficulties are solved (although there are few conflicts that you should crack earlier). Good!

Interestingness

The plot starts with some tension.

The story reaches a nice climax with a good amount of tension. This is an important characteristic of a good narrative. Great!

A better end would contribute to having a more interesting tale.

There are surprising events that make the story appealing. I enjoyed that!

Novelty

I find this story pretty original! I love it!

My evaluation of your story is 100/100

Dangerous

by Pablo Gervás and Carlos León,
Universidad Complutense de Madrid

Russian tales, virtual worlds, dangerous curves... There are lots of sources of inspiration for programs that write stories! Pablo Gervás and Carlos León tell us about some of the ways their team at the Universidad Complutense de Madrid are exploring. How long before you are watching a Hollywood film where the storyline is credited to a program?

The shape of stories to come

One way to devise an algorithm to write new stories is to look at the structure of existing stories. People have studied literature for centuries, and have had some useful ideas about the shape stories take. Vladimir Propp studied Russian fairy tales and decided that they all fit a basic pattern: a Hero sets off on a journey to undo a wrong perpetrated by a Villain, they may be aided by a magical Helper, and they are rewarded at the end by marrying a Princess. Sound familiar? Based on Propp's ideas, our team have developed a program called Propper that can create stories by following this pattern. To make sure the stories make sense it knows about the things that must be true before particular actions can happen – a villain must have a sword if he is to stab a prince, for example. It also knows what the consequences of those actions are – that prince will be hurt if stabbed! Although Russian folk tales were used as the original inspiration, we're now adding more story structures. No doubt you can think of lots more. The detective who has to solve a crime committed by a murderer hiding in plain sight? The man in the street who saves the Earth from an alien invasion after the authorities fail? By extending Propper, and collecting enough patterns, we hope to build a reasonably good storyteller: one that can at least match the formulaic plots of your run-of-the-mill Hollywood movie!

Curves

Every story under the sun

Structure isn't the only way to produce a good story. When you play a video game the story changes based on what you do.

Our team started with this idea as a way of creating stories. Our system, STellA, simulates a virtual world, just as video games do. In a virtual world lots of things can happen based on the rules of the world: characters eat, sleep, talk, have hobbies, and so on. As the world is simulated actions happen and a story unfolds. STellA is a bit different from a normal virtual world though. It generates all events and situations that could possibly happen within the limits of that world. It's as though a human writer worked out all possible storylines for their story: like Shakespeare writing out every feasible ending for Romeo and Juliet. While it isn't possible for a human writer, computers can do it in seconds. Why write one story when you can write them all!

Don't forget your readers!

Whichever way you write stories, it's important to remember they are written to be read by someone. Your stories will work better if when writing them you keep in mind what the reader is going to be thinking at each point. "Have I already told them that the old lady had a gun in her purse?" "If I say this now, will it give away the ending too soon?" Writers think about questions like that over and over as they write. They then rewrite their drafts based on the answer, adding, deleting or modifying things until the worries go away. Another line of research being followed by our team is to model this kind of behavior by writers so that we can use it to improve our whole family of storytelling programs.

Our model captures the way in which the writer (or program) working on a story does these things repeatedly: drafting, reading, asking questions about the draft, and redrafting until all their worries have been resolved. The story is then told or written down in its final version. Most existing storytelling programs (including ours) currently cover only a small part of the cycle that human writers follow. Once we extend them to include the missing operations, the stories they produce will improve massively.

Put all the different approaches together in one program and Hollywood watch out! What a story that will make!

The curve of a story

Unfortunately just generating lots of stories isn't good enough. One of the most studied problems in Computational Creativity is how to decide on the best result; the most creative work from those generated. STellA has to tackle this problem too: if you produce lots of simulations, you have to decide which one gives the story you want people to read. STellA deals with it in two ways. The first is to let the human who runs the program set conditions for the resulting story: they might require that the hero doesn't marry the princess at the end, for example.

The other way is to use curves! You pick some aspect of the story, like the amount of emotion involved or degree of danger, and represent it as a number. Suppose you chose danger. As the versions of the story progress the degree of danger in each changes in different ways and so the number representing it goes up and down in different ways too. Now think of that number changing as plotting a graph – the story is being modeled as a curve that goes up and down as the danger changes. A way to chose a story from the many possible ones simulated is to pick the one that best matches a given shape of curve. You might for example choose a story curve where the danger is high at the start, drops to nothing, then builds rapidly to the end. That would give a different story to one where its danger curve was always low. Based on its 'narrative' knowledge of the kind of curve that makes a good story, STellA can choose the best stories generated from its multiple world simulations.

Composing from Compression

by Geraint Wiggins, Queen Mary University of London

Computers compress files to save space. But it also allows them to create music!

Music is special. It's one of the things, like language, that makes us human, separating us from animals. It's also special as art, because it doesn't exist as an object in the world— it depends on human memory. "But what about CDs? They're objects in the world", you might say and you'd be right, but the CD is not the music. The CD contains data files of numbers. Those numbers are translated by electronics into the movements in a loudspeaker, to create sound waves. Even the sound waves aren't music! They only become music when a human hears them, because understanding music is about noticing repetition, variation and development in its structure. That's why songs have verses and choruses: so we can find a starting point to understand their structure. In fact, we're so good at understanding musical structure, we don't even notice we're doing it. What's more, music affects us emotionally: we get excited (using the same chemicals that get us excited when we're in love or ready to flee danger) when we hear the anthem section of a trance track, or recognise the big theme returning at the end of a symphony.

Surprisingly, brains seem to understand musical structure in a way that's like the algorithms computer scientists use to compress data. It's better to store data compressed than uncompressed, because it takes less storage space. We think that's why brains do it too.

Sound waves only become music when a human hears them

Even more surprisingly, brains also seem to be able to learn the best way to store compressed music data. Computers use bits as their basic storage unit, but we can make groups of bits work like other things (numbers, words, pictures, angry birds...); brains seem to do something similar. For example, pitch (high vs. low notes) in sequence is an important part of music: we build melodies by lining up notes of different pitch one after the other. As we learn to hear music (starting before birth, and continuing throughout life), we learn to remember pitch in ever more efficient ways, giving our compression algorithms better and better chances to compress well. And so we remember music better.

Our team use compression algorithms to understand how music works in the human mind. We have discovered that, when our programs compress music, they can sometimes predict musical structures, even if neither they nor a human have "heard" them before. To compress something, you find large sections of repeated data and replace each with a label saying "this is one of those". It's like labelling a book with its title: if you've read Lord of the Rings, when I say the title you know what I mean without me telling the story.

If we do this to the internal structure of music, there are little repetitions everywhere, and the order that they appear is what makes up the music's structure.

If we compress music, but then decompress it in a different way, we can get a new piece of music in a similar style or genre. We have evidence that human composers do that too!

What our programs are doing is learning to create new music. There's a long way to go before they produce music you'll want to dance to—but we're getting there!

Music-making mates for Mortimer

by Louis McCallum, Queen Mary University of London

Robots are cool. Fact. But can they keep you interested for more than a short time? Over months? Years even?

Roboticists (thats what we're called) have found it hard to keep humans engaged with robots once the novelty wears off. They're either too simple and boring, or promise too much and disappoint. So, at Queen Mary University of London we've built a robot called Mortimer that can not only play the drums, but also listen to humans play the piano and jam along. He can talk (a bit) and smile too. We hope people will build long term relationships with him through the power of music.

Robots have been part of our lives for a long time, but we rarely see them. They've been building our cars and assembling circuit boards in factories, not dealing with humans directly. Designing robots to have social interactions is a completely different challenge that involves engineering and artificial intelligence, but also psychology and cognitive science. Should a robot be polite? How long and accurate should a robot's memory be? What type of voice should it have and how near should it get to you?

It turns out that making a robot interact like a human is tricky, even the slightest errors make people feel weird. Just getting a robot to speak naturally and understand what we're saying is far from easy. And if we could, would we get bored of them asking the same questions every day?

Would we believe their concern if they asked how we were feeling?

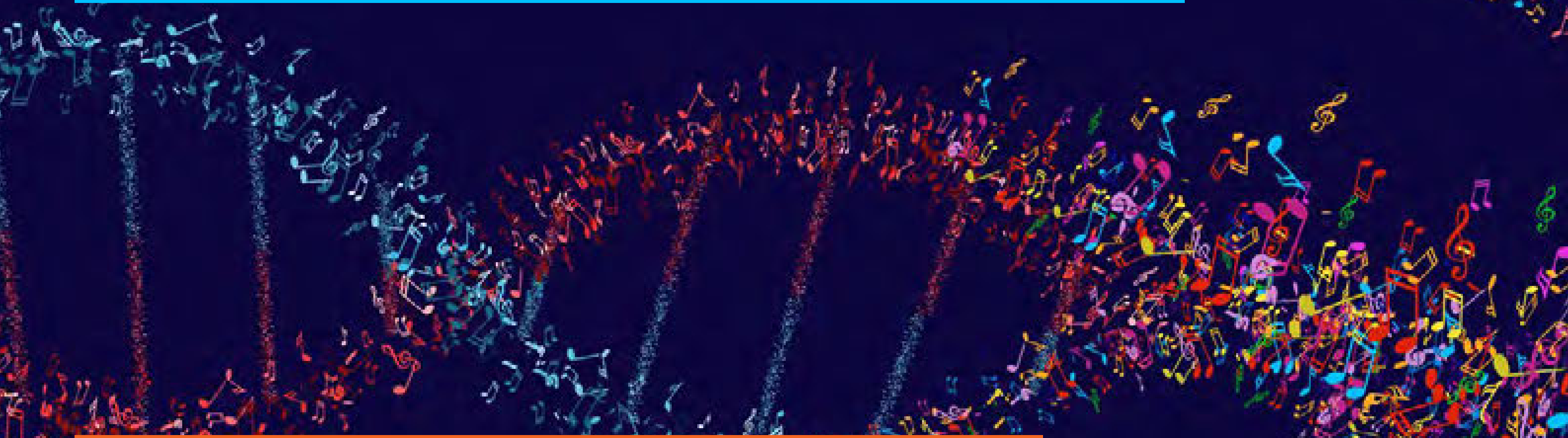
Music is emotionally engaging but in a way that doesn't seem fake or forced. It also changes constantly as we learn new skills and try new ideas. Although there have been many examples of family bands, duetting couples, and band members who were definitely not friends, we think there are lots of similarities between our relationships with people we play music with and 'voluntary non-kin social relationships' (as robotocists call them - 'friendships' to most people!). In fact, we have found that people get the same confidence boosting reassurance and guidance from friends as they do from people they play music with.

So, even if we are engaged with a machine, is it enough? People might spend lots of time playing with a guitar or drum machine but is this a social relationship? We tested whether people would treat Mortimer differently if it was presented as a robot you could socially interact with or simply as a clever music machine. We found people played for longer uninterrupted and stopped the robot whilst it was playing less often if they thought you could socially interact with it. They also spent more time looking at the robot when not playing and less time looking at the piano when playing. We think this shows they were not only engaged with playing music together but also treating him in a social manner, rather than just as a machine. In fact, just because he had a face, people talked to Mortimer even though they'd been told he couldn't hear or understand them!

So, if you want to start a relationship with a creative robot, perhaps you should learn to play an instrument!

Music that has sex

by Paul Curzon, Queen Mary University of London



Music is a creative part of our culture but what drives that creativity? Music changes over time as new sounds are invented and old ones fall from favour. Where does the creativity come from though? Is it from the 'creative people': the musicians? Or is it you and me that really drive the creative changes, just by the music we choose to listen to? Could a machine make creative new music we love without an actual composer? Researchers based in London and Japan teamed up to investigate. They did it by creating a Darwinian music engine – a musical machine that has sex!

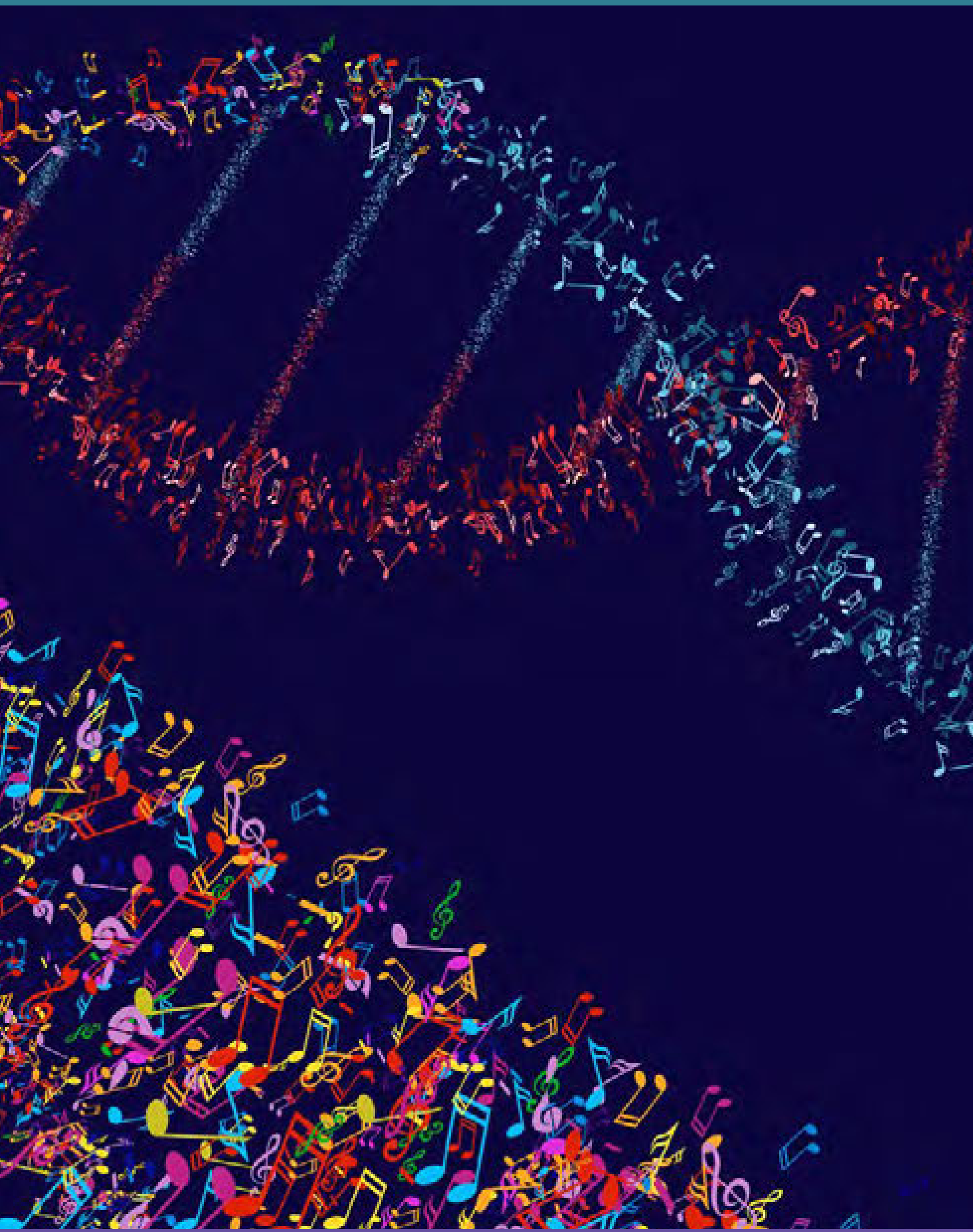
Well ok. It doesn't really have sex, but it does do a software equivalent. Life evolved through natural selection and sex is central to that. When we have a baby, we are mixing up the genes of the parents giving that child qualities from both. Your DNA is a complex code storing all the information about how to make you. DNA determines the colour of your eyes, how many legs you have, even whether you have a shell or not! It is unique to you – no living thing is exactly the same as you so (twins aside), nothing else has exactly the same DNA to describe them. Sex is just a way to split two creatures' DNA in half and pair up the two halves to make new DNA and so a new creature.

Natural selection then works because the fittest creatures, most able to survive the conditions they find themselves born into, survive to have babies of their own and so pass their DNA on. That DNA contains information about what made them special enough to do so well, increasing the chances that the baby does well too.

You can model this in a computer. First create a code to represent the properties of the thing you want to evolve – its 'genome'. It can just be a string of numbers, one for each property. Next, create a random set of these strings to start things off. Then the sex begins. Pairs of strings are split in half and joined back together. Extra random changes to the strings add mutation to the process. Test the newly created population, keeping the best and destroying the worst. For that you need a 'fitness' function to decide what is good and bad. In nature the fitness function is your ability not to die. In the software version it can be anything that captures your idea of good and bad.

The researchers investigating musical culture, did this: their 'creatures' were bits of sound. Their artificial 'DNA' – the strings of numbers – represented different music. Each encoded a computer program. When that program is executed it plays a short, seamlessly looping sound sequence. The genome/program determines things like where notes are placed and instrumentation, though other things like the tempo are identical for every loop. Sexual combination and mutation mimic the fusion of existing musical motifs, rhythms, and harmonies, and the invention of novel ones.

Their music engine, 'DarwinTunes', started with a group of short audio loops



Breeding art

Evolution has been used to create art by a Japanese team. In their system paintings breed by doing things like chopping two pictures in half then re-pairing the halves. The test to see if a picture survives is based on preferences set at the start by a person indicating the style of art they like. The program then runs for thousands of generations judging the results against the preferences, keeping those that fit best for the next round of breeding. The system is based on the observation that paintings by human artists follow a similar pattern of using features of existing art in new ways rather than completely inventing new ideas. The researchers think you could set up the preferences to match the style of any artist creating new 'Rembrandts' or 'Monets' by starting with their existing paintings. Would the result be a new painting by that artist? **They think so!**

that played random noise. These loops paired up, sexually reproduced and mutated, creating new loops of music. They were left to evolve over 2,500 generations of musical 'creatures' with daughters replacing their parents on each round. The twist was that the selection was based on the likes and dislikes of thousands of people who rated the music clips for how much they liked them. Only the top 100 survived in any round.

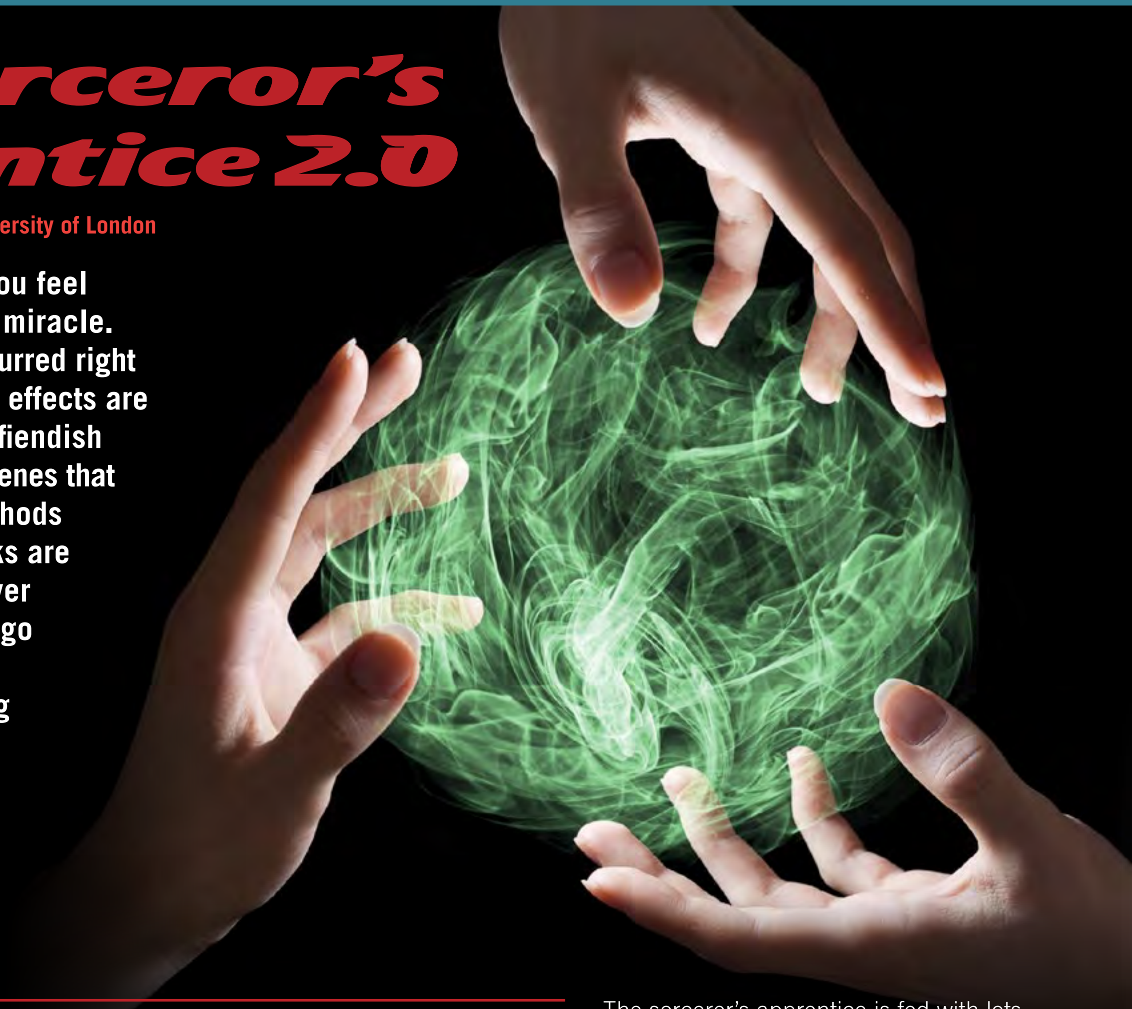
> **Their 'creatures' were bits of sound**

The loops of random sounds quickly evolved into music. This was partly because pleasing chords and rhythms used in western music started to evolve. Later, however, the amount of evolution slowed and there was little improvement after 600 odd generations. This pattern of fast then slow evolution is actually seen in the real world: in the wild, the fossil record and in lab experiments. To work out what was going on with the music the team carried out other experiments using methods devised by biologists studying the evolution of bacteria. It turned out the slow-down was mostly because of a decrease in the accuracy of how the music was transmitted. A similar thing arises in early musical cultures when, as musicians learn existing complex musical themes they make mistakes so the original themes are lost. Once the evolution got to a certain point, favoured but complicated innovations were being lost, so improvements could no longer build well on those that came before.

The main aim of the experiment was to understand how musical culture develops. It also shows though that by using a Darwinian process machines can make pleasing music without a composer. However there is more to it than just natural selection – to be as creative as human composers driving musical cultural change something else is needed. It is more than just sex and mutation! Creative machines will need some other spark that composers have.

The Sorceror's Apprentice 2.0

by Howard Williams, Queen Mary University of London



A good magic trick makes you feel like you've just witnessed a miracle. An impossible event has occurred right in front of you. Most magical effects are simple to experience, but a fiendish complexity lies behind the scenes that is hard to work out. The methods behind some of the best tricks are so complex no-one would ever believe the magician would go to such enormous lengths to pull off such a simple seeming trick. Magicians will do almost anything to make you gasp in wonder, and go to similar lengths to prevent you finding out how they did it!

The ultimate sorcerer's apprentice: a computer program that suggests new magic tricks

Magicians constantly look for new ways to wow an audience. They often head the queue to try out new technology, and often invent things to use in tricks that go on to be used elsewhere. Magicians were involved in the birth of cinema, exploiting and refining the ways film could be edited and manipulated to create magical movies. Today's blockbuster special effects are the end product of this on-screen conjuring.

At Queen Mary University of London we've been turning our coding skills towards making new magic tricks using Artificial Intelligence techniques from our labs. Artificial Intelligence (AI to us geeks!) isn't just a Steven Spielberg movie, but a whole field of Computer Science. It's dedicated to finding ways to make computers intelligent: to program computers to give them ever more sophisticated ways to solve problems and discover new information from huge amounts of data. Soon, artificial intelligences will be driving cars on our streets (they already fly planes in our skies!), cleaning our houses, running our home's heating systems, possibly even being our friends or work colleagues!

With a little help from in-the-know magicians, who have revealed some of their arcane secret methods, our research team have created what could one day be the ultimate sorcerer's apprentice: a computer program that suggests new magic tricks.

Computers are far better than humans at doing sums incredibly quickly, and storing huge amounts of information without ever forgetting it. As a result, they are exceptionally good at picking out patterns where, to the human brain, there don't seem to be any. They can direct all their number crunching abilities at complex problems that would otherwise take humans years or even centuries to complete. For example, computers are currently used in medicine to help scientists understand how DNA works. They sift through the billions of ways in which different bits of DNA interact to cause all sorts of changes in human bodies. These same pattern finding abilities of AIs can be used to sift through the various ways to build magic tricks and find the ones that work the best.

The sorcerer's apprentice is fed with lots of information about how we perceive the world. Based on that information it churns out new magical methods, leading to new tricks, that should amaze an audience in the best way possible. It's a program that is able to find the very best version of a trick at the click of a button!

Amongst the recent tricks it has come up with are a magical jigsaw puzzle, and an astounding card trick during which a mobile phone reads the mind of a spectator. All the cunning of a magician's mind is needed to know what will fool real people – but it takes a clever AI to figure out the ultimate way to really confound them. When performed well, the tricks really do leave the spectator thinking a miracle has occurred. We know this because, being scientists, we tested the tricks out to see how mystifying they were! Result: very!

We may prefer to have an actual person perform the magic for us (though robot magicians are just around the corner too), but what magician wouldn't want a handy AI assistant around to help them craft their next masterpiece? The audience need never know!

The rise and fall of the living dead!

by Peter W. McOwan, Queen Mary University of London

Zombies seem to be everywhere these days and if playful fighting against the undead is your hobby, there is an app for that. Created by Queen Mary University of London student David Kilanowski, Trapdoor Zombies adds creative twists to the classic zombie chase plus some floor dropping gameplay for good measure. There is a twist in the way the game was created - a creative game-playing Artificial Intelligence (AI) helped.

The artificial intelligence program creates endless waves of (not too intelligent) zombies moving through level maps of increasing difficulty. The cunning layout of each is machine-designed and tested to ensure it is playable. Peppered throughout the levels are trapdoors. If your thump your phone, the motion sensor opens the trapdoors and drops anything unlucky enough to be on top to their doom below, be it zombies or you. This fast paced, fun strategy game sees you race against the clock to shoot the spawning hordes, drop the zombies through the trapdoors and release in-game characters to build your zombie busting team. To make sure the AI was producing usable designs David experimented with the many variables that control the gameplay and difficulty, testing each till he found the right mix. Once they were in the code the machine was ready to create.

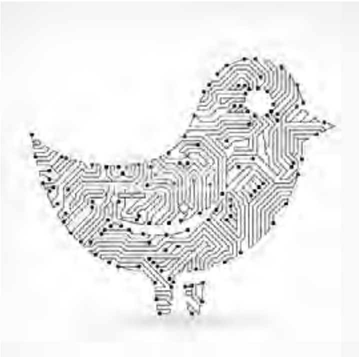
If you fancy taking on the undead or just seeing how AIs can design games, download the app at **http://www.qappsonline.com/apps/trapdoorzombies/**

Magic through the trapdoor

You take a deck of cards, shuffled by a spectator, and deal them into two piles. Three cards are selected from one pile, and placed face up. Various face down cards are dealt on to these face up cards. You use the values and suits of the three face up cards to correctly predict a card and its position in the face down pile. If your spectator tries to repeat your amazing feat they fail. It's a trick that can't be explained! FInd out more at **www.cs4fn.org.**

Tours of Babel

by Tony Veale of University College Dublin



When you wish big, be careful what you wish for. Take a story called The Library of Babel by Argentine writer Jorge Luis Borges. The library of the story is a fantastically vast construction of hexagonal rooms, containing every book that was ever written, as well as every book that can, or will, ever be written. The library is finite but huge: it contains every book that can be written with a 25 character alphabet in 410 pages at 80 characters per line and 40 lines per page. It's easy to imagine such a library, and easy in principle to construct one too, but the library is not as useful as it might seem. For every good book hiding on the shelves, there are countless millions of nonsense books containing random text, and worse, countless millions of half-nonsense books that mix real insight with random nonsense. There is no definitive catalogue for the library, but since a catalogue is itself just a book, the library contains millions of books claiming to be catalogues, almost all of which are gibberish, while many are just misleading.

Borges showed us that extremes are easy: to generate everything we just need an alphabet and a simple way to combine them, and to generate nothing we need just do nothing. The real challenge is to generate something in between with meaning and literary merit that is not random and not a rehash of what someone else has already written.

Simple ways combining letters or methods that cut up existing texts and put them back together in random orders are called mere generation, generation for the sake of it. Though the results might occasionally be seen as creative by an outside person, the generator itself can't tell the good outputs from the nonsense, just like Borges' library has no way of separating the good books from the bad. What if we start with an existing book by a respected author, and make scripted changes to this book to obtain a new one that is different but meaningful? We call this pastiche if the new text remains close enough to the original to successfully piggyback on its meaning. When we build software systems to be creative like humans, it is tempting to build ones that specialize in mere generation or in pastiche, because these two extremes are the easiest to implement. They also need the least amount of knowledge about the world to be built in to the software.

To see mere generation and pastiche in the wild, just go to Twitter: some of the biggest offenders are human, relying on the same old tropes and cut'n'paste techniques, but there are plenty of computers too, in the guise of Twitterbots. Some of these 'bots use mere generation and pastiche to humorous effect, such as @pentametron, which puts together pairs of random tweets from real people if

they can be given a poetic cast in iambic pentameter. Here's an example: "Still waiting for the good in that goodbye" & "It's really fun forgetting to reply." The trick is that the reader is willing to imbue this pair of lines with the connective tissue of real meaning, even if the lines are only chosen for their obvious rhyme.

Computational Creativity researchers are now building Twitterbots that use their knowledge of the world to generate micro-texts that the bots themselves consider meaningful and worth sending out into the world. This new generation of bots must apply their own filters about meaning and beauty to figure out what they are trying to say and to determine how well they are saying it, throwing away the worst and only tweeting the best. A good example of this new kind of bot is @MetaphorMagnet, which generates a novel but meaningful metaphor every hour or so. Here is an example: "To some amnesiacs, memory is a treasured blessing. To others, it is an overlooked error. #MemoryOrBlessing" How might you build a metaphor machine that generates new but meaningful metaphors? It's easy to build a machine that generates random combinations of words that fit a template like "An X is a Y", but how would you choose X and Y to offer a valuable perspective that can help people to think about a familiar topic in apt new ways? If you have some good ideas, why not put them into action with a Twitterbot of your own?

One day the majority of users of Twitter may be bots, bouncing texts and ideas off each other in a magnificent society of artificial minds, as we humans look on in wonder and amusement.

Back (page) on the block

by Peter W. McOwan, Queen Mary University of London

Blocks, acting as a verb, to block, get in the way (of creativity). As a noun, a block, they are things we use to create stuff. That's the thing about blocks, you're never quite sure what you're going to get. Here we take a Computer Science look at the ways this block bimodality behaves.

Block on the landscape

Fancy visiting Denmark? You can without leaving the comfort of your computer thanks to the Danish government. They uploaded a 1:1 map of Denmark into Minecraft, where every Danish detail is recreated by millions of tiny computer graphics blocks. Rather than millions of hours of game play to build the landscape, the full sized replica was created using geographical map data. The upload was a hefty Terabyte of block building data – a billion or so blocks of 1s and 0s. The Danish government hope players will enjoy wandering the land, adding new buildings and places. The only rules: no swearing, no bullying no use of virtual TNT!

Big benefit: Big bulks of blocks bring bi-location benefits

Build-a-block

Architects are taking blocks to a whole new level. 3D printing is used to build prototypes of designs. The printer moves, controlled by a computer, depositing blobs of materials that stick together in layers, building up the desired shape. But think bigger. What if the printer was the size of a cargo container and deposited blocks of material to create a full sized house? That's what they are doing in Amsterdam! Building with bricks is so yesterday! In China waste materials are recycled to create blocks to print a cheap bungalow in a day or so. Others are thinking cuddly, looking at how modified 3D printers can join together blocks of knitting for making soft toys.

Big benefit: Blocks are bricking it and building buddies

Block calculations

The brain, like a computer, is a powerful information processor. To do the monumental calculations needed to stay alive, a brain has different blocks for different abilities. Your eyeballs are the only part of your brain you can touch. They are at the front of the brain but the process of seeing starts to happen at the back. Information from your eyes runs through brain areas called, simply, V1 to V5. Damage area V3 and you can't perceive colour. Damage V5 and you may not be able to see movement. Your ability to produce fluent conversation seems to happen at the side in a place called Broca's area: problems there cause difficulties with speech. Dyscalculia, a rare condition where the patient can't do arithmetic also seems to have particular brain areas associated with it. We can explore the brain's processing of information in these areas using clever algorithms, giving us a better understanding of how we compute, or don't, in our heads.

Big benefit: Broken blocks of brain beckon mental blocks

Block bends beat behaviour

Cities like New York are built on a grid. Streets run in parallel creating city blocks, which make it easier to get around. But not all cities are so simple to navigate and that's where maps come in. Where we are changes the things we see and feel, but what if it could also change the music we're listening to? Enter Geosound, an app that uses the map of where you are to resequence the music on your phone. Select a track and the app calculates its beat structure, and then downloads a map of the streets around you. It extracts the roads and junctions of your location and uses them to jump the music through its beat structure while showing how the beat changing elements are moving around you on the map. If your location gives an exciting rework of a musical classic you can tweet it for others to experience. Download it from: **www.qappsonline.com/apps/geosound/**

Big benefit: Blocks and beats make mash-up music

cs4fn is edited by Paul Curzon, Jo Brodie and Peter W. McOwan of Queen Mary University of London. Ursula Martin, Bernard Sufrin and Peter Millican of Oxford University gave valuable input. This issue and/or the research in it has been supported by the projects: CONACYT-México, project number 181561, WHIM (611560), Lrn2Cr8 (610859), ConCreTe (611733), PROSECCO (600653) and COINVENT (611553) funded by the European Commission, Framework Program 7, the ICT theme, and the Future and Emerging Technologies FET program. It has also been supported by the Department of Education, Mayor of London through Teaching London Computing and Google. Autumn 2014.

Department for Education | EPSRC Engineering and Physical Sciences Research Council | Queen Mary University of London | SUPPORTED BY MAYOR OF LONDON | COMPUTING AT SCHOOL EDUCATE · ENGAGE · ENCOURAGE