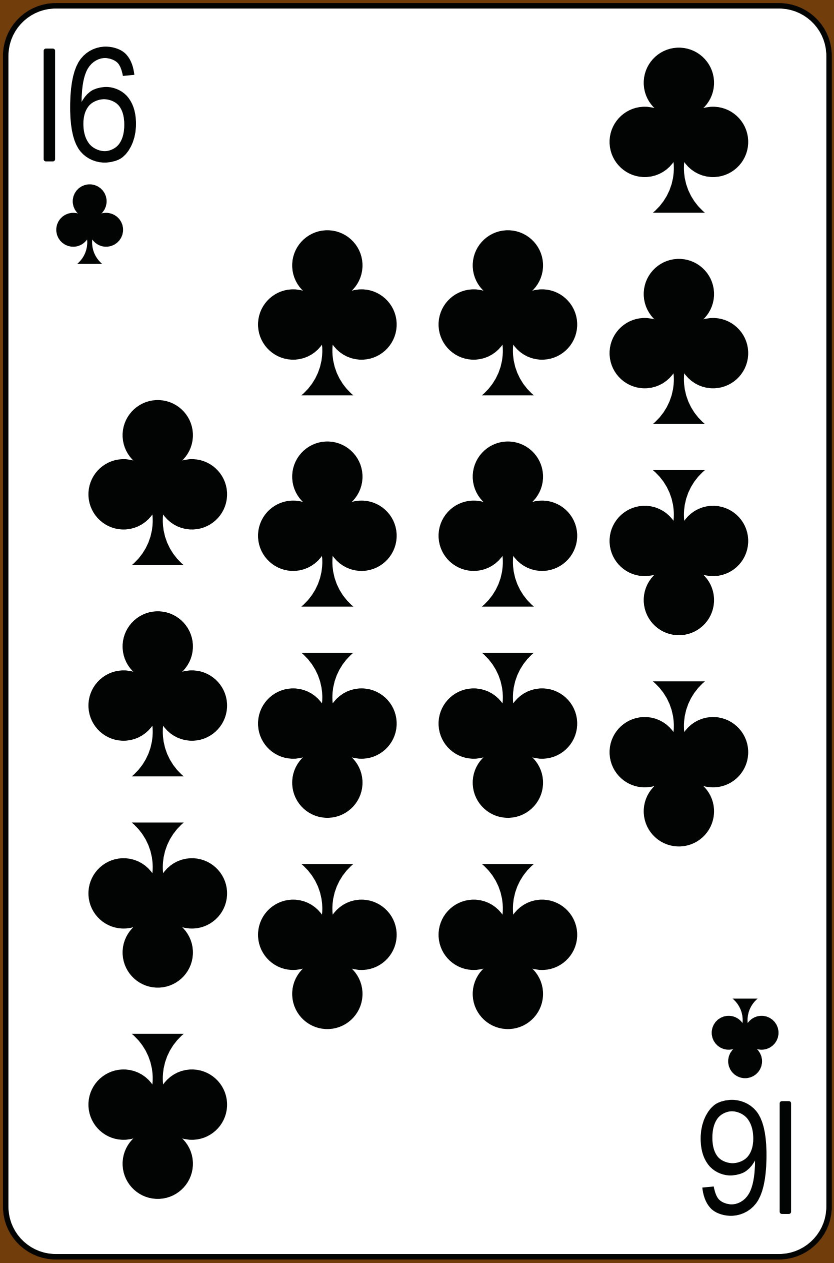
# The Magic of Computer Science II

## Now we have your attention...

A medley of magnificently magical marvels mischievously manipulating mind mistakes

**Presented by**
Peter McOwan, Paul Curzon and Jonathan Black of the School of Electronic Engineering and Computer Science, Queen Mary, University of London

Queen Mary
University of London



This is your chosen card

# Contents

# Conjuring, cognition and computing

Magic is about entertainment. It is also about psychology, mathematics and computer science too.

To be a good magician you need to know more than just the secret of how a trick works. Great magicians also have a flair for cognitive psychology: they have a natural understanding of people. They manipulate where you are looking and what you see, what you forget and what you remember... and that includes making you remember things that didn't actually happen.

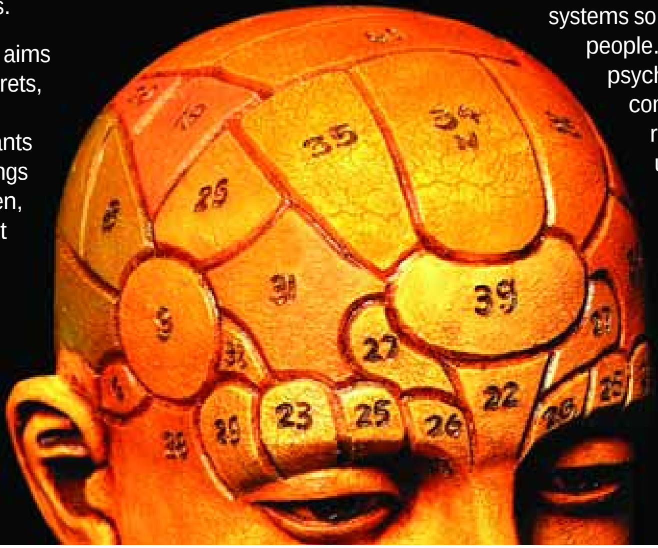
Scientists are exploring these same issues. They talk about studying attention, memory, cognition and perception. Their experiments manipulate these things to work out how the brain works.

A magician aims to keep secrets, whereas a scientist wants to bring things into the open, and present them for all to see.

Research scientists aim to work out precisely what is causing the effects observed rather than just watching, and to preserve this knowledge for future generations.

Science is about working out how the world (or even the universe!) works. Understanding how our brains work is perhaps one of the most fascinating areas right now. It turns out things we take for granted about what we see and remember are not always as straightforward as we think.

What does this have to do with computing? Well, computer scientists use those same scientific results to create better computer systems, using their knowledge to help make lives better. Human-computer interaction (HCI) is an important area of computer science. It is the study of how to design computer systems so that they work for people. Just as magic draws on psychology, so does human-computer interaction. The rules that psychologists uncover are turned into design principles for program designers to follow, allowing wizard computer scientists to conjure up programs to change our world.



# How to use this book

## Magic

This booklet contains a series of tricks. We describe the magical effect that you are aiming to create. We then describe its mechanics – what you do to make it work. We also give a proof the trick does work. After all you don't want it to go wrong when you perform it for an audience. Finally, each trick comes with a showmanship section. It suggests alternative ways to present the trick that may give that all-important extra WOW factor. You can also experiment with your own variations once you know the core secret that makes the trick work.

## Computer science

Every trick has links to some fundamental aspect of computer science. The second section of each trick describes these computer science links. The idea is that as you learn about the magic you will also learn some interesting science, maths or engineering too. In particular we include a link to human-computer interaction (HCI), explaining the psychology behind both the HCI and the trick.

## Keep the magician's code

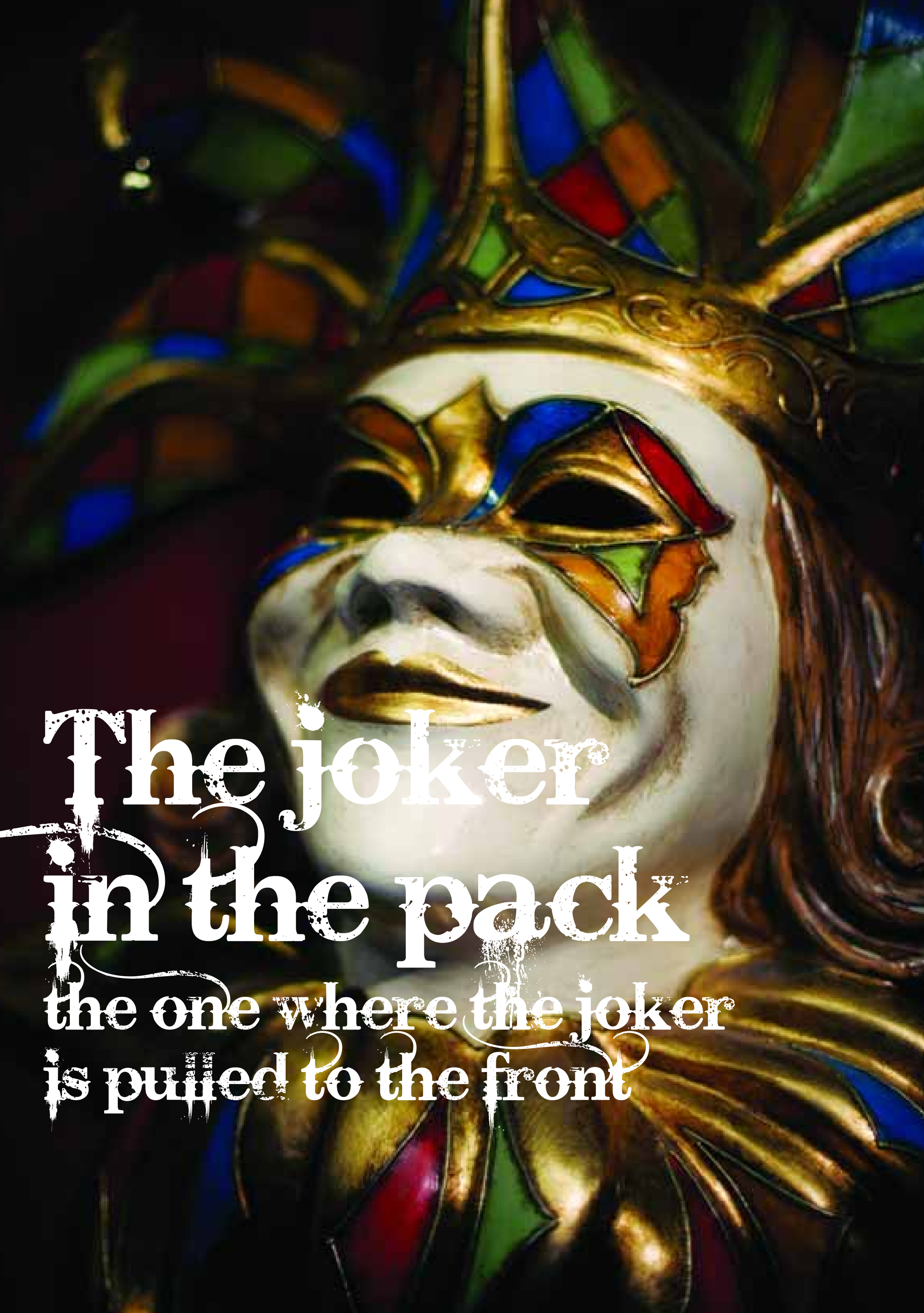
Some of these effects are actually in the shows of professional magicians. We present them here for educational and entertainment purposes. If you do perform them for friends don't break the magicians' code. Never reveal the secrets to your audience.

# Keep the secrets!

## It's in the presentation

Keep the secrets, yes, but also remember that in magic, practice and presentation is just as important as the secret. Ultimately it is the final effect on your audience that matters. With a slightly better presentation a trick that last week had a mediocre reception can suddenly cause gasps from your audience.

Computer programming is similar to magic in this way. Programs combine the code to do the job (like the secret) with a user interface (like the presentation) through which the human user interacts with the program. Just as with a trick, the same program can be transformed from unusable junk to a best seller by getting the interface right. iPhones are one recent example of the effect a revolutionary interface can have on sales. Revolutionary magic can just as easily be down to the presentation too.



# The joker in the pack

## the one where the joker is pulled to the front

# The joker in the pack: the one where the joker is pulled to the front

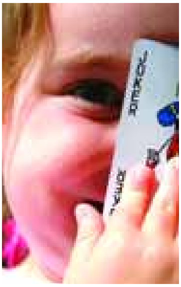
## The magic effect

You take an ordinary pack of cards and ask for a volunteer from the audience. Remind them that teachers must be able to spot troublemakers – jokers – in their class and get them to the front before the trouble starts. You are going to apply a test to see if the person is a natural born teacher.

You ask them to think of a small number and tell you what it is. They count out that number of cards on to the table. You ask them to turn over the top card and to their surprise they have found the joker and brought it to the front straightaway.

## The mechanics

This trick is very simple. Just place the joker at the top of the pack before you start. Get them to pick the number (let's say they choose 5). As you explain to them that you want them to count that many (5) cards on to the table, you show them how. Now pick the 5 dealt cards back up and place them on top of the pack, giving the pack to the volunteer. Get them to count the cards out themselves dealing them on to the table: 1, 2, 3, 4, 5. They turn over the last card and magically it is the joker.



## Prove it works!

The joker is at the top of the pack. Whatever number the person says, you deal that many cards on to the table. If they picked 5, then there are now 5 cards on the table. The joker is at the bottom, in the 5th position. Placing these cards back on top of the pack leaves the joker as the 5th card. That is exactly where it needs to be so that when the volunteer deals 5 cards themselves it is on top of the pile on the table.

Suppose they picked some other number. The same reasoning applies. If they pick the number n, then the joker moves to the nth position in the pack.  Dealing out the first n cards pulls it to the top of those dealt.

## The showmanship

Don't draw too much attention to the fact that you have dealt out the cards. That is the part that makes the trick work, so you need to do it casually: you are just making sure the volunteer understands what to do.

It is also important to put in a delay between each deal of the cards and before the joker is revealed. Think of some patter for these bits. Quickly take back the remaining pack as they finish dealing. Fan the cards and point out they are a just an ordinary pack of cards. They aren't all jokers! Perhaps pick out some individual cards to show them. This draws attention to the pack and what you are doing with your hands and away from the dealt cards. It also means they are more likely to forget precisely what happened earlier. Magicians call this time misdirection, and it's a useful psychological tool. Try it out and work out what forms of misdirection work for you.

# The computer science

# The joker in the pack: the computer science

## Algorithms

Magicians call a trick like this one, that is sure to work without the need for sleight of hand, a self-working trick. A computer scientist would call it an algorithm.

An algorithm is just a series of instructions that if followed guarantees some effect. The algorithm tells you what to do and what order to do it in. Why is this important to computer science? Well, computers cannot think for themselves. All they do is blindly follow instructions: an algorithm. A computer program is just an algorithm written in a language that a computer can execute.

When writing out an algorithm you have to specify the order that things are done. The simplest way of doing this is using **sequencing**. Here is our joker trick written out as a sequence of instructions. The instructions have to be followed in the order given.

**1** In private place the joker at the top of the pack.

**2** Select a volunteer.

**3** Ask them to pick a small number.

**4** Explain they must count that many cards from the pack.

**5** Show them how by counting the number of cards on to the table.

**6** Add those cards back to the deck.

**7** Ask the volunteer to count the cards out themselves.

**8** Ask them to turn over the top card revealing that it is the joker.

Follow this algorithm, executing the instructions in the given order, and you can be sure the joker will appear on top at the end of the trick.

## Human-computer interaction

If you do the trick following the algorithm, it will work technically. The joker is guaranteed to end up at the top. That doesn't mean it will work as a trick though. The audience might just be left confused or see through it straight away.

For example, suppose you didn't tell the audience just before the reveal that the point was to find the joker. Why should they be impressed that it is that card? It had to be something! The algorithm worked but the presentation let it down. Even though it is not needed to get the joker to the top, a story like the one about teachers helps make the trick work as it gives the trick meaning.

Similarly, if humans are to use programs, then the presentation that sits on top of the algorithm has to be good. The **human-computer interface** that acts as the intermediary between the program and the user must work too. What is the interface? It is just more program, more instructions. In our trick the part in the middle is:

**1** Quickly, take the remaining cards from the person.

**2** Spread the pack so all cards are visible.

**3** Pick out a couple of cards to highlight and name.

When describing the trick, we could have written the showmanship instructions mixed in with the rest. We separated them out as that

# The joker in the pack: the computer science

makes both parts easier to understand. There are similar advantages to splitting off the core instructions and the interface parts of a program too. It makes the program much easier to write and different members of a team can work on the different parts. Also, programs are often changed after they have been written, as customers want them to be able to do new things. It is much easier to do that if the program is split up with each part doing a clear thing.

## iPhone magic

Magic effect designer Andrew Mayne has produced some free magic stunts that use the graphics capabilities of iPhones. Find out more at **www.cs4fn.org/magic/**

## Human error

One way your presentation might let you down is if you do not include enough delay after the cards are dealt. If the audience see the two deals together or know immediately where the joker has ended up, they can work backwards remembering what they just saw and so where the joker must have started. By adding a delay and giving them other things to think about, they are less likely to remember what happened.

Many psychologists think in terms of two kinds of memory. **Long-term memory** is where things are stored that you have actively memorised. **Working memory** is where you store things that you are currently processing. Imagine watching a vote-for-the-celebrity TV show. When you remember your favourite's telephone number long enough to dial the number you are using your working memory. You have a very limited amount of working memory. That is why long phone numbers can be hard to remember. People can only hold around seven chunks of information in working memory at a time, like seven digits of a phone number. It's not quite that simple though. If you organise the digits into groups each can become a chunk.

As an interface designer, you have to take into account human fallibility. People make mistakes and some kinds are more likely than others because they depend on working memory. Take car headlights, for example. Have you ever seen a driver get out of a car having forgotten to turn them off?

What was going on? They switched the lights on so knew they were on. They knew they had to switch them off on arrival. They were relying on their working memory to remember, but in the meantime there was a long delay while they did all the driving. Driving involves concentrating on lots of other things that fill up working memory. Eventually they arrived so had achieved their goal. They were probably thinking about what they had to do there. The little tidying-up task of switching off the lights went out the window!

This kind of error is called a **post-completion error.** They occur when a tidying up task must be done *after* you have achieved your goal. Ever gone to find something in your room and forgot to turn the light off when you left? Your goal was to get the thing you were after. Turning off the light is just a tidy-up task. Experiments have shown that post-completion errors happen when a person's working memory is filled.
If you have lots of other things to remember at the same time you are more likely to make the mistake.

## Design principle

***Design to avoid human error.***

The over-arching design principles for interface designers here is to design to prevent people making these kinds of likely mistakes.

There are lots of ways a designer can help prevent post-completion errors happening. One way is to redesign the order things are done so that the goal is only achieved after the tidy-up tasks have been done. UK cashpoints return your card before they give you cash for exactly this reason. Another way is to give **cues**. They are just reminders that something needs to be done. For example, modern cars often beep if the driver's door is opened and the headlights are still on.

Cues are not fool proof though. What kinds of cues work best? Experiments have shown that to be effective in preventing post-completion errors, cues have to be just-in-time. The cue has to appear just at the point when you need to do the action. If the car only reminded you before you reached the destination, for example, then you would still frequently forget.

Back in the magical world, if you were to remind the audience just before the joker is revealed that you had dealt out the cards, you would ruin the magic. Instead help them remember the things that will confuse them: that the volunteer had a free choice and that they dealt that number of cards.

# Are you psychic?

## the one where you test the psychic powers of a member of the audience

# Are you psychic? the one where you test the psychic powers of a member of the audience

## The magic effect

You test the abilities of a spectator who believes they have no psychic powers. You give them a pile of ten cards and tell them that they're going to try to make a pair without looking at any of them. The cards are separated into two piles and then the spectator mixes them up a little under your direction. You discard the top two cards on each pile. You work through the piles that way, mixing and discarding until there are only two cards left. To the volunteer's astonishment, the last two cards are a pair. They are psychic. As their jaw drops you show them that they are even better than they thought. It turns out every eliminated pair of cards match in value too.

## The mechanics

Before you start, get the ace to the five of hearts and the ace to five of spades out of the pack. Set them up in one pile in the order red 1-5 cards then black 1-5. Spread the cards in your hand and have the spectator point to the back of any card. Split the pack at that point and place the top pile at the bottom of the pack. Repeat this until they are happy the cards are well mixed.

Deal the top five cards onto the table, reversing their order. Place the remaining un-dealt cards in a second pile beside them. This keeps their order the same.

You explain that as there are 5 cards in each pile you will give them 4 moves to test their psychic powers. A move involves taking the top card on one of the piles and placing it on the bottom of the same pile. They can, for example, do all 4 moves on one pile, 2 on each, or 3 on one and 1 on the other. It is their choice, their intuition, remembering that their aim is to be left with two matching cards.

Once the 4 moves are made, remove the top card on each pile and place them aside, face down. Point out that it does not matter what they are because they are being discarded.

Now there are 4 cards in each pile. Offer the spectator 3 moves, make those 3 swaps and then remove the top two cards from the piles. Keep doing this giving them one less move than the number of cards. Eventually you are left with two single cards. They will match. So will all the other pairs discarded.

# Are you psychic? the one where you test the psychic powers of a member of the audience

## Prove it works!

Believe it or not, this trick works automatically. Let's see why.

At the start we have the first pile of cards in order 1234512345. Cutting these cards with a single cut doesn't disturb the cyclic order of the cards. So, for example, a cut between 2 and 3 may move the card order to 3451234512. This means if you were simply to split the 10 cards in half, the two sets of 5 cards are in the same order in both halves. It's just the order is now 34512.

For simplicity, let's assume that the cards are in the original 1234512345 order. Our secret move, counting five cards and reversing their order, gives us 12345 in one pile and 54321 in the other. This is called a **palindromic stack**. The values in one pile are exactly reversed in the other. To make it even simpler to explain, consider only 3 card values so our piles are 123 and 321.

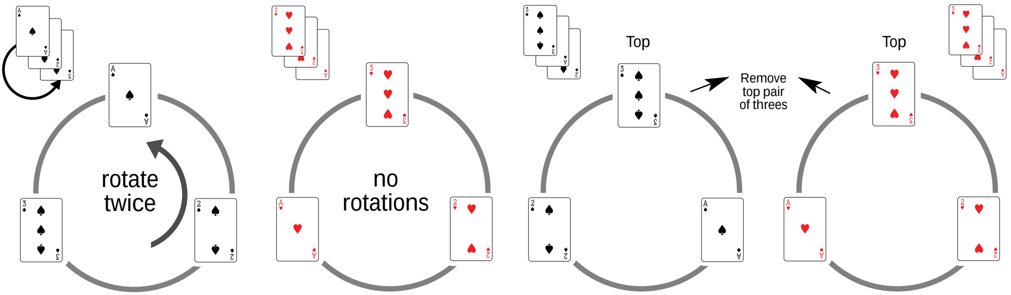
Now make 2 moves. Remember the number of moves is always one less than the number of cards in the piles. Moves just rotate the cards. Think of the cards in positions on a rotating clock face. Start with 12 o'clock being the top card and the other two cards being at 4 o'clock and 8 o'clock. See the diagrams at the bottom of these two pages. We rotate the dial by the number of moves. Rotating one less than the number of cards (i.e. twice in our 3 card example) just means we have shifted the bottom card up to the top: to 12 o'clock. That is the card that starts at the top of the other pile. If we do some of the moves on one pile and some on the other, we are just rotating the numbers on the dials in opposite directions so that they meet. You can work through all the possibilities to see how it happens. The mathematics behind this trick is called modular arithmetic...not surprisingly also known as clock arithmetic.

This mathematics works for any number of cards n that are placed in order in palindromic stacks. When we make n-1 swaps, as one card is moved down, its equivalent rises in the other stack so that the ones meeting on top of the piles match. After removing the top two cards we still have palindromic stacks so we can continue.

**Moving one pile** →

Start (2 moves total)

Final position after first pile has both moves



## Showmanship

You can play up the psychic link by using Zener cards (buy them from a magic shop or make your own). Each has one of a circle, cross, wavy lines, square or star on them. That hides the fact that they start in order. They were invented by psychologist Karl Zener as a way to do rigorous experiments on whether extra-sensory perception exists. Typically, the experimenter turns the cards over, writing the symbol down, and the participant tries to work out, using clairvoyance, what the card was. Sometimes to remove the possibility of bias or subtle forms of communication, the participant is in a different room.

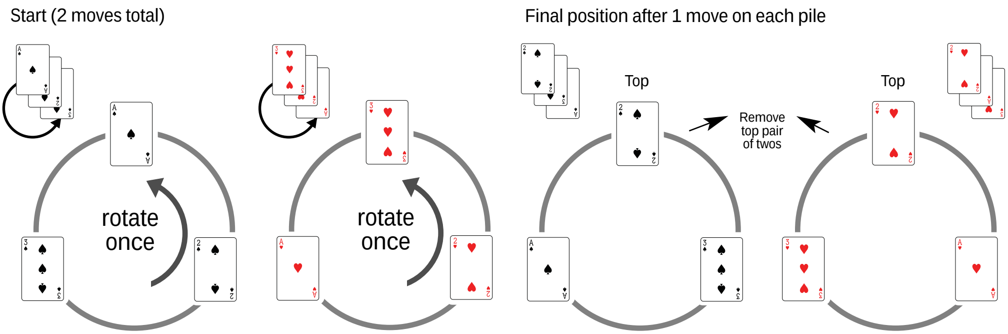
As part of your patter, explain this. Add that you will do a variation of the Zener experiment. The extra detail adds authenticity to your performance, casting you in the role of experimenter rather than of magician, and again gives the trick meaning and an expectation of strange things happening.

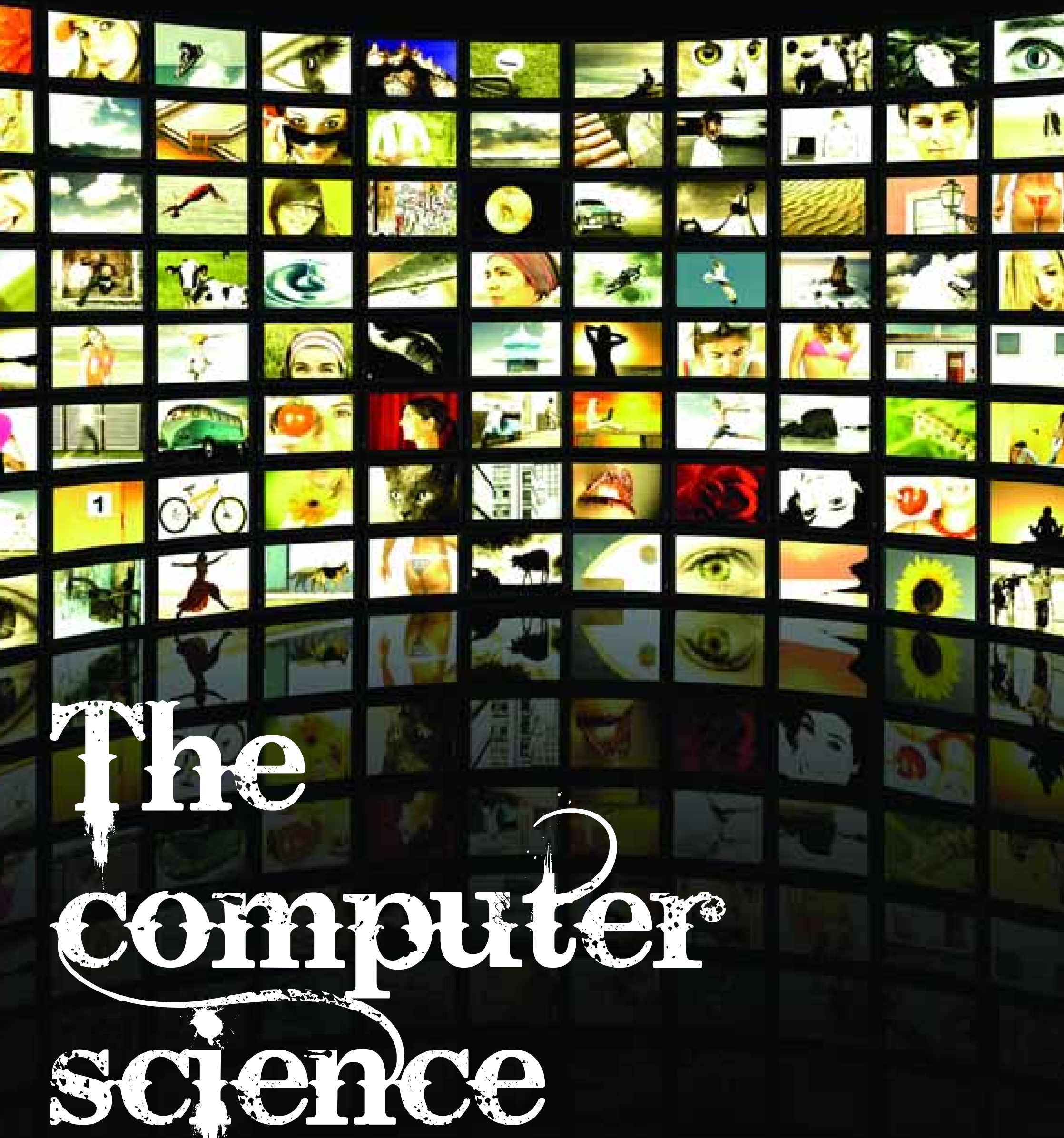
When you get to the last two cards in each pile, point out that this is the final chance to get it right. One move is left, and one card can make all the difference!

Alternatively, cast the trick as a demonstration of ‘psychic resonance and the power of crystals’. Give the volunteer a large ‘crystal’ (e.g. a piece of quartz) to hold to ‘magnify their psychic resonance’.

In the film Ghostbusters one of the Ghostbusters, Peter Venkman, uses Zener cards to experiment on how electric shock punishment affects psychic powers.

**Moving both piles**





# The computer science

# Are you psychic? the computer science

## Algorithms

This trick always works if you correctly follow the steps: it's another algorithm. Algorithms do not have to be made of simple steps that just follow one after the other. They can include loops too. A **loop** is just an instruction that says: "repeat the following steps some number of times". Below are the mechanics written out in a more program-like way. It is still a list of instructions following one after the other, but this time some are done over and over.

### The psychic algorithm

```
1. Order the cards:1,2,3,4,5 of hearts then
   1,2,3,4,5 of spades.
2. Spread the cards, cut them, then deal out
   the top 5 cards.
3. Place the rest on the table.
4. Set a counter (that we will call n) to be 4.
5. Repeat the following until n is 1:
    5.1 Ask the person to pick a number (k) that is
        between 0 and n.
    5.2 Move k cards to the bottom of the first pile.
    5.3 Move n-k cards to the bottom of the
        second pile.
    5.4 Put the top card of each pile, as a pair, to
        one side.
    5.5 Subtract 1 from the counter n.
6. Reveal that all pairs match.
```

To write this out as an algorithm, we used *n* and *k* to keep track of numbers that we needed to remember for a later step. They are called **variables**. A variable can hold different numbers at different times.

*n* is a variable that keeps track of how many times we've repeated the instructions and so also how many pairs still have to be removed.

*k* is a place-holder for the number of moves the person wants to make on the first pile. We need it immediately to do the moves. We also have to subtract *k* from *n* to get the right number of moves for the other pile. Programs use similar operations on variables, moving numbers between variables, doing calculations and storing answers in new variables.

# Are you psychic? the computer science

## Human-computer interaction

We've seen that interface designers use their skill to make interactive systems easy to use. This is called **usability**. There is more to human-computer interaction than usability though. Another important thing for designers to think about is user experience. It's about how you design systems to be enjoyable to use or engaging, for example. It's really important if you are designing a game or an interactive museum exhibit. It can also be important for more mundane things like an online shopping site. Think of the opposite to having a good user experience. Imagine a site that was irritating, for example. That would definitely ruin a business. If customers are going to remember a site and keep coming back, the site should give them a sense of delight. Online shopping is often mundane but it doesn't have to be!

Delight is something a good magician knows all about. The magician wants you to have an enjoyable time but if they can delight you then you will come back again. Tricks like this psychic trick do that. You set up an expectation that the volunteer will match the last cards. It delivers that. The audience are already impressed, their expectation met. However, when the trick appears to be over, it delivers more. Then you reveal that not just one pair matches, they all do. Astounding! The audience are delighted.

Magicians often think of a great trick as having three parts:

**The Pledge** – you set the scene, giving the effect its meaning

**The Turn** – you perform the trick leading them along with the story

**The Prestige** – the reveal of the twist at the end, the delight!

Can you identify these three stages in each of the tricks described in this book?

## Design Principle

***Design the experience not just the interface.***

Computer scientists are starting to explore user experience and things like how to engineer delight. Games programmers are way ahead, of course. Delight is an arms-race: eventually the things you were doing to create delight will become normal. You have to do more. You need to be more creative than the competition. The same goes for magicians. That is why they are always endeavouring to do newer, more impressive tricks, always trying for the unexpected. User experience is everything.

## Magic in space?

Computer games designer, magician and scientist Richard Garriott bought a ride to the International Space Station in 2008 as a space tourist. Using the magic of computer science he was able to 'read minds' from the space station and performed the first ever micro-gravity magic show in space.

Psychology and audience expectation play a big part in magic. Magicians cleverly use science to apparently break the laws of physics, but what about when you perform where the laws of physics are different? Richard created his effects with this in mind. In space tennis balls float freely so the magic is in making them fall to your feet! How did he do that?

# Mass hypnosis

## the one where you force everyone to think of the same card

# Mass hypnosis: the one where you force everyone to think of the same card

## The magic effect

You explain that all great magicians can make a volunteer pick a card of the magician's choosing. You are going to try something much harder. You are going to use your powers of suggestion to make as many of the audience as possible pick the card you choose, all at once. You show five playing cards on a screen. The audience focus on the first card their eyes settle on and then look at no others. Once they have picked a card they raise their hand. Once all the hands are up, you blank the screen and tell them to keep concentrating – they must remember their card and theirs alone. You explain that you will then deal out the cards again but you will have removed one card. It is the card you think most people will have settled on. Once you show the cards, anyone who chose the card that is gone should put their hand down. You show the cards again and all hands go down down. You made them all pick the right card!

## The mechanics

This trick is simple. You just swap *all* the cards for new ones. As none of the cards will remain it doesn't matter what cards anyone chose. To be effective you need to use very similar, nondescript cards: 7s, 8s and 9s of similar red and black suits. That ensures the differences in the cards they didn't choose are not very noticeable.

## Prove it works!

Trivial. They all change so of course everyone should put their hand down.

## Showmanship

To increase the effect you need to give the audience an expectation of what you are about to do. Claim that you are going to try an experiment on the power of suggestion. Play up the point that what you will do requires a deep understanding of psychology and the ability to make subtle suggestions. Point out that as there are five cards, from chance alone, one in five people will sit down. For the experiment to work you will need to make far more people than that choose your card.

# The computer science

# Mass hypnosis: the computer science

## Computer vision

This trick relies on people not noticing changes from the original cards to similar cards. It works because all things in an image are not equal, and some things we see are more noticeable than others. That may be an interesting psychological phenomena, but why are computer scientists interested? Computer scientist Milan Verma certainly is. His PhD work at Queen Mary is investigating how to design a computer vision system that 'sees' like a human.

## Interesting!

A human-like vision system, if accurate, should agree with people about what things in a scene are **salient:** which parts attract the eye. That is what Milan's computer model does. It identifies how interesting each part of an image is based on things like the relative brightness of bits of the image.

Many different things affect what is salient. For the cards in our trick, the other values are not cognitively salient: their differences are hard to notice because the numbers are similar ones, and they weren't ones you were focusing on. Also the new cards look very similar to the ones you took away, so that affects what you notice too. It is this latter visual salience that Milan's program tries to model.

Having developed a vision model, how do you test if it really does see like a human? Milan wrote a program that generates patterns to test his model against human perception. Each is made up of random looking shapes. The twist is that inside each pattern is a square that seems to pop out when a person looks at it. The square just looks out of place against the background. The more obviously it pops out, the more 'interesting' it is taken to be.

By combining the vision model with the pattern generation program Milan created a test program. It works using ideas based on Darwin's natural selection (see 'Breeding salience' on page 25). You give the test program a number to act as a 'salience value' and it creates a pop-out pattern that is exactly that obvious to the vision model. The bigger the value the more 'interesting' the result should be and so the more obvious the pop-out should be when the model 'looks' at it.

Milan then showed the patterns to humans to see if they agreed with the computer model's predictions. Get agreement and the model works!

Milan has shown that humans respond exactly the way the computer predicts: the computer model is a good model of the human vision system. That suggests we have a good understanding of how our brain decides visual salience. You can try it yourself with the patterns on page 25.

## Brush strokes

Once we have a good vision model we can then start to use it. Milan has created another program that paints pictures. It takes a photo and applies brush strokes to make a painting effect. However, more care is taken in the detail of the 'interesting' parts of the picture, leaving course brush strokes in the less interesting parts. How does it decide which parts are interesting? It uses his model of visual saliency of course!

Once you have an eye for what is interesting you can not only do magic but create art too!

## Design principle

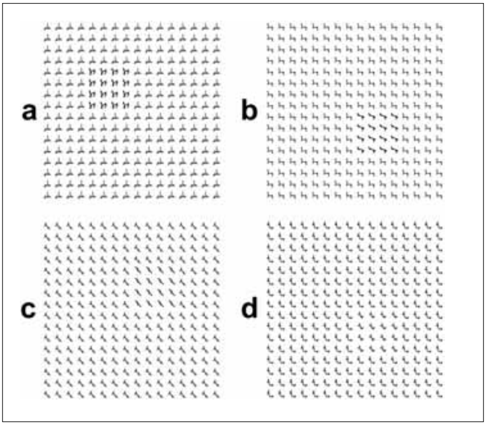
***Make important changes visually salient!***

Interface designers control what appears on a computer screen, whether on a DVD's menu screen or in the control panel for an entire city's metro. They must design the elements so people notice the important things. It's obvious that warnings must be salient, but other more innocuous things must be too.

Take a medical infusion pump – the hospital devices that control how much drug a person gets. They are used by the nurses in surgery to control the amount of anaesthetic the patient gets. It's really important that the nurse notices if they type in a dose of 27 rather than .27, for example, or the patient gets an overdose. That might happen if they didn't press the decimal point button hard enough. That small decimal point needs to be very salient. So does the fact that it isn't there! Someone's life might depend on it.

If something changes on the screen (or sometimes if it doesn't change) and it is important, the change had better be salient!

**Five magicians co-authored a paper in a top scientific journal in 2008 on 'Attention and awareness in stage magic'. It looks at how understanding magic tricks can help understand the human brain.**



Can you spot the difference? The computer artificial intelligence says a) is easy, b) is a bit harder, c) is harder still, and d) is very tricky. What does your brain say? If you aren't sure where the squares are check your answers at **www.cs4fn.org/magic/**

## Breeding salience

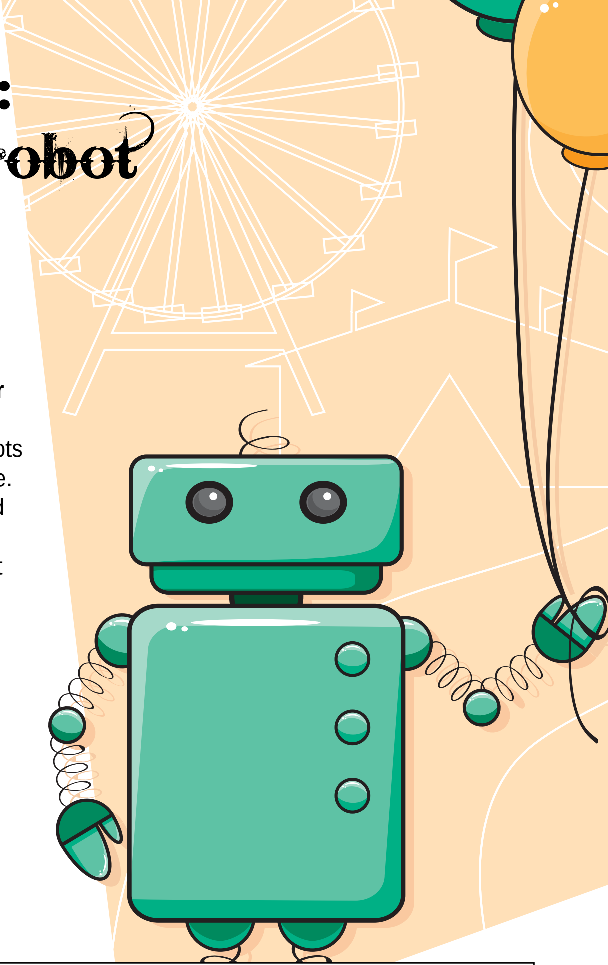
Milan's work is the first time that artificial intelligence has been used to create patterns with such precise differences in salience to test human perception. His program uses a form of artificial intelligence called a genetic algorithm. It is a process based on Darwin's ideas of natural selection and the 'survival of the fittest' that drives evolution.

The artificial intelligence first generates a set of random patterns. In nature, 'survival' means having children before you die. For a test pattern the survival test is about how close the model rates it to the given target value of salience. Patterns survive if, from all those currently competing, they have values judged closest to the target. The others are killed off. New patterns (children) are created by making random changes (mutations) to the survivor patterns. The children are then tested for fitness and the process continues.

Keep doing this over and over and all the small but successful mutations build up, all the while getting closer to the desired level of salience. The result is a box that pops out exactly the right amount.

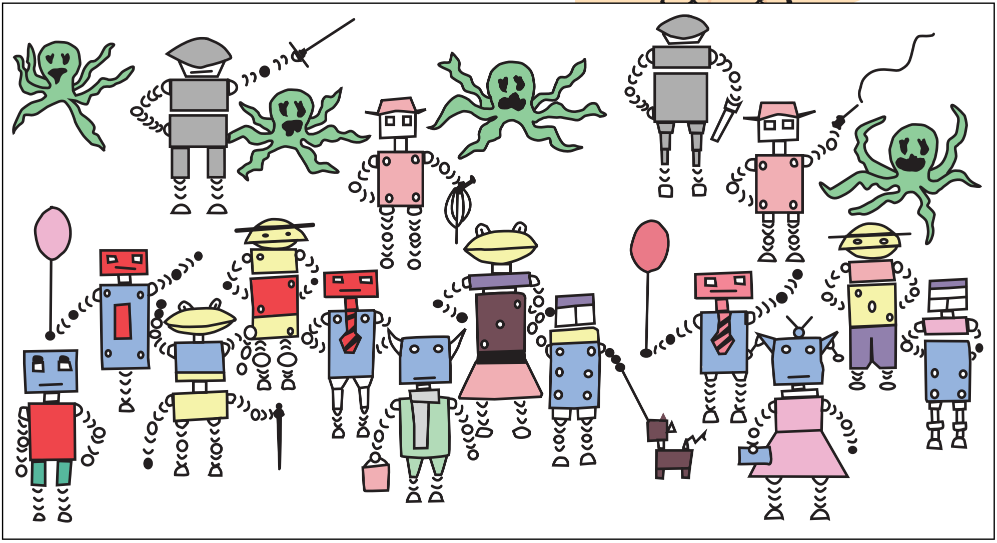
The artificial intelligence breeds never-before-seen patterns to a custom-set level of difficulty. For some images it's easy to spot the difference. In other images it is much harder.

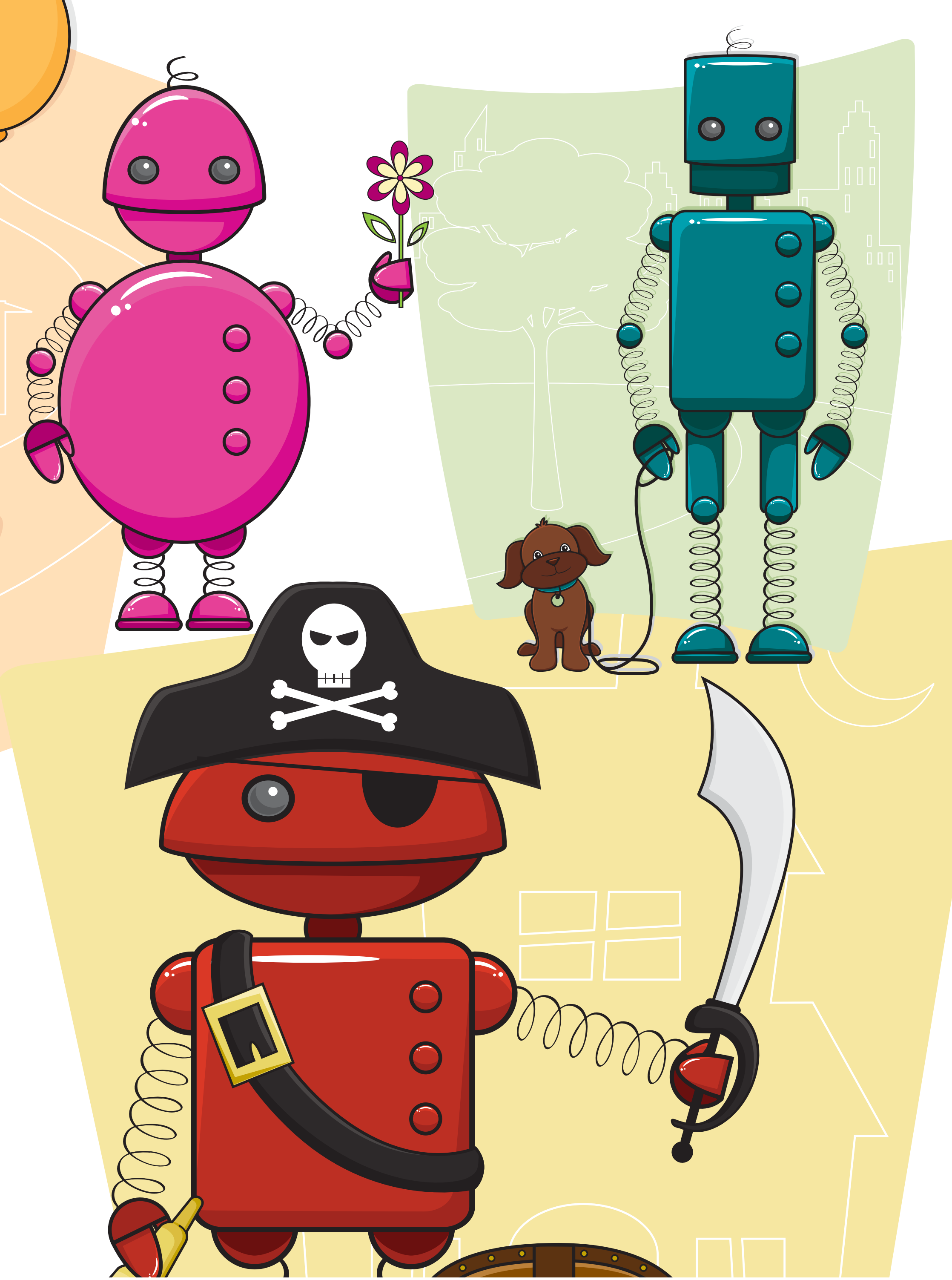
# Just a quick one: the teleporting robot



**The one where the robot disappears before your eyes.**
You introduce a simple jigsaw showing 17 robots in a row. Your audience count them one by one. You collect up the four pieces of the jigsaw and concentrate, then reassemble the pieces. As you count the robots again, the amazement spreads through your audience. It's the same jigsaw as before but only 16 robots remain. Number 17 has vanished, leaving no trace!

Get a large copy of the jigsaw from **www.cs4fn.org/magic/**. Find out how to make the robot teleport while learning more about human perception and secrets behind some big stage tricks.





# The mind-boggling memory experiment

## the one where you memorise the whole pack

# The mind-boggling memory experiment: the one where you memorise the whole pack

## The magic effect

After years of training, world memory champions can memorise packs of shuffled cards and recall them in perfect order. In this experiment you memorise half a pack.

A spectator shuffles the pack then divides it in half and gives you one half. Quickly scan them saying that you are memorising the pack then place them in your pocket.

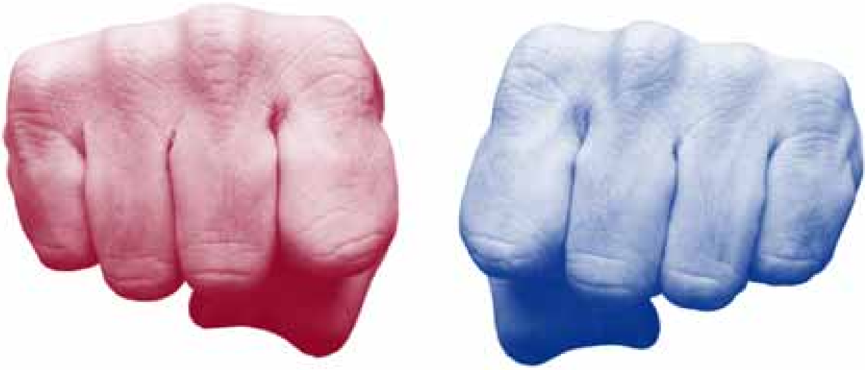
Another volunteer names any card. Suppose they name the ten of diamonds. You think carefully, and state that the ten of diamonds isn't in your pocket. However, you say you can still prove you memorised the cards. First you put your hand in your pocket to try and match the named suit. "Hmm, 12th card from the top is the eight of diamonds I seem to remember," you say. You pull out the eight of diamonds, just as you said. You've matched their suit with a diamond.

"Of course," you continue, "there was a one in four chance of getting a diamond at random, so I'll pull out another card to make the value of your chosen card, the ten of diamonds". Without much effort you remember there was a two of hearts "five cards from the bottom" and pull it out: eight of diamonds and two of hearts. 8+2 =10 as required, proof you have amazing skill to memorise cards in an instant!

### Just a quick one: the blended bodies experiment

In this effect you prove you can remotely sense the world through another person's body. To demonstrate this remarkable skill you hand the spectator a coin, and turn your back. Without looking at them instruct the spectator to make a fist with each hand and hide the coin in one of their fists. To let you begin to blend your body with their body, ask them to raise the hand with the coin in it and hold it to their forehead and to concentrate on how the coin feels as you start to resonate with them. After a short time you tell them you are ready, and that they should hold both fists out straight in front of them. You turn around for the first time, and as your body is now tuned with theirs you are correctly able to identify the hand concealing the coin.

**Is this a case of blending bodies or bending truth?**
Find out at www.cs4fn.org/magic/



# The mind-boggling memory experiment: The one where you memorise the whole pack

## The mechanics

Secretly take the ace of clubs, two of hearts, four of spades and eight of diamonds from the pack. These are your secret 'stack' and you need to memorise them, but that's easy! For the suits, remember CHaSeD. The capitals represent the order of the suits – clubs, hearts, spades and diamonds.

Hide these 4 cards in your pocket beforehand. In the trick, put the half pack in your pocket so that the four hidden cards go on top.

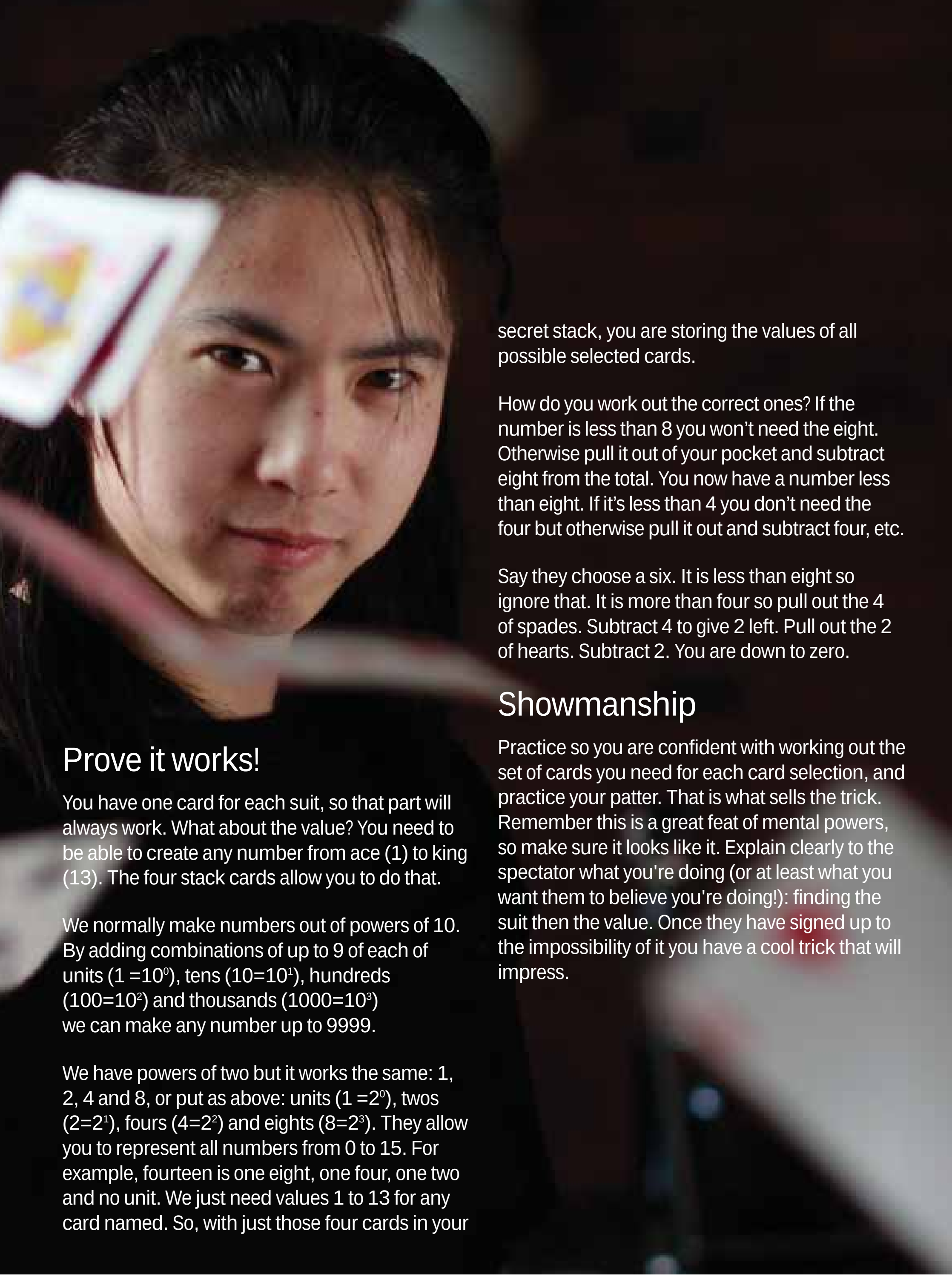
When they name a card, one of two things can happen. They could mention one of your four memorised cards. If so you act like you are remembering where in the pack it is and then pull it out from amongst the top 4 cards…to stunning effect! But what if they don't mention one of your secret cards?

The first part is easy. For any card named you can pull out the secret card with the same suit. Just remember CHaSeD.

For the value you pull out those cards from the four that add up to the right number – ace (1) through 10, jack (11), queen (12), king (13). Include the suit card in the addition if you need it.

So if the card selected is the king of clubs, bring out the suit first – the ace of clubs. Then for the sum pull out the 8 of diamonds and the 4 of spades to add to the ace on the table, 1+4+8=13, the value of a king.

If they say the ace of diamonds, pull out the eight of diamonds first (suit), then the ace of clubs (value). Here you don't add in the suit card to make the addition.



## Prove it works!

You have one card for each suit, so that part will always work. What about the value? You need to be able to create any number from ace (1) to king (13). The four stack cards allow you to do that.

We normally make numbers out of powers of 10. By adding combinations of up to 9 of each of units ($1 = 10^0$), tens ($10 = 10^1$), hundreds ($100 = 10^2$) and thousands ($1000 = 10^3$) we can make any number up to 9999.

We have powers of two but it works the same: 1, 2, 4 and 8, or put as above: units ($1 = 2^0$), twos ($2 = 2^1$), fours ($4 = 2^2$) and eights ($8 = 2^3$). They allow you to represent all numbers from 0 to 15. For example, fourteen is one eight, one four, one two and no unit. We just need values 1 to 13 for any card named. So, with just those four cards in your secret stack, you are storing the values of all possible selected cards.

How do you work out the correct ones? If the number is less than 8 you won't need the eight. Otherwise pull it out of your pocket and subtract eight from the total. You now have a number less than eight. If it's less than 4 you don't need the four but otherwise pull it out and subtract four, etc.

Say they choose a six. It is less than eight so ignore that. It is more than four so pull out the 4 of spades. Subtract 4 to give 2 left. Pull out the 2 of hearts. Subtract 2. You are down to zero.

## Showmanship

Practice so you are confident with working out the set of cards you need for each card selection, and practice your patter. That is what sells the trick. Remember this is a great feat of mental powers, so make sure it looks like it. Explain clearly to the spectator what you're doing (or at least what you want them to believe you're doing!): finding the suit then the value. Once they have signed up to the impossibility of it you have a cool trick that will impress.

# The computer science

# The mind-boggling memory experiment: the computer science

The trick relies on being able to represent numbers by adding the powers of two: the binary representation. That is exactly how numbers are stored in a computer. Computer memory is like a vast bank of switches. Each can be on or off: position 0 or position 1. So to store something in memory we need to code it up as a sequence of 1s and 0s. That is what binary does for numbers. Fourteen was 1 eight, 1 four, 1 two and 0 unit: that's 1110 in binary. We can't use decimal with its digits 0 to 9 in a computer as the memory switches are 2-way not 10-way.

## Human-computer interaction

Magicians do more than manipulate where the audience's attention is focused to ensure we miss the things that give a trick away. They also give us other things to think about. In the 'joker in the pack' trick we saw that people are more likely to make some kinds of mistake if their working memory load is filled. Experiments have shown that the story is more subtle than that though and it can make a trick go wrong too.

We fill a person's working memory by imposing a **cognitive load** on them. One way this can happen is that the task they are trying to do could be difficult in itself. The load is a central part of what they are trying to do. This kind of load is called an **intrinsic load.**

The magician has that problem with this 'memory experiment' trick. To pull out the right cards you have to do some arithmetic in your head. The intrinsic load is high so you are more likely to forget to do steps in the trick. That's what makes it hard, not that you have lots to memorise!

Cognitive load could instead be just due to the way information is presented. This is called an **extraneous load.** If the information you need to do the task is hidden amongst lots of other information then there is an extraneous load.

For example, suppose in another trick you have to watch as cards were turned from the top of the pack to see when a card appears. (The 'You said it!' trick on page 35 is like this.) That monitoring is adding an extraneous load.

The extra patter that a magician gives around a trick is partly there to add extraneous load. The information the audience need to work out what happened is there. The trouble is they have to filter it from all the other things the magician is saying and doing. If everything is stripped down to the core steps for the mechanics to work, then tricks are much more obvious.

Why do we care that there are different kinds of loads? Well it turns out that they affect different cues in different ways. We have discussed visual cues: cues that you see, like a reminder message. There are also internal cues. Whereas visual cues are out in the world, **internal cues** are in your head. As you practice a series of actions over and over again each starts to cue the next one. You apparently don't need visual cues of what comes next any more.

Jonathan Back of University College London has explored how cues are affected by different kinds of loads. His experimental participants role-played doing fire-engine dispatch. They had to respond to incidents, plot routes on a map for fire engines to take and allocate back-up fire engines, all using a computer interface.

# The mind-boggling memory experiment: the computer science

For some participants the possible routes were plotted automatically (giving a low intrinsic load as the task itself was then quite easy) whereas others had to work out the best route themselves (a high intrinsic load: a hard task).

All had to extract information that affected the possible routes (like traffic jams) from a streaming ticker tape of information. For some, much of this was irrelevant to the current route (high extraneous load). Others were just given the information they needed in the ticker tape (low extraneous load).

There were lots of opportunities for people to make errors whilst dispatching fire engines. Only three were common. The first was an initialisation error where a person forgot to do a setup step before doing the main parts of the task. Forgetting to click in a password box before you type is an example of this kind of error. A second was a post-completion error where people missed a final tidying up task (remember the mistake of forgetting to switch off your light on getting something from your room). Finally a mode error was also common. The error happened when people failed to check a mode indicator that told them what they needed to do next. Was the machine in a state where they would be given route possibilities or was it in a state where they had to work out the route themselves? They assumed the wrong mode so did the wrong thing.

The experiments showed that when the task was hard and there was lots of information to sift through  (high intrinsic and extraneous load), visual cues were far less effective and people were much more likely to miss them. They were more likely to make mistakes.

What does that mean to you in a performance? If a trick needs you both to do mental calculations and to watch cards being dealt for the values needed, don't rely on noticing subtle reminders from your assistant to stop you forgetting a step!

You might expect that internal cues would be affected in the same way. That isn't true. It turns out that all you need is a high intrinsic load for internal cues to be disrupted, and so make it more likely that people will make errors. In the memorising trick you do have a high intrinsic load, so beware. In pressure situations like a performance you will be prone to forgetting things however much you practice. If you have to do tricky stuff in a trick, keep the rest simple!

## Design principles

***If tasks are hard make important cues highly salient.***

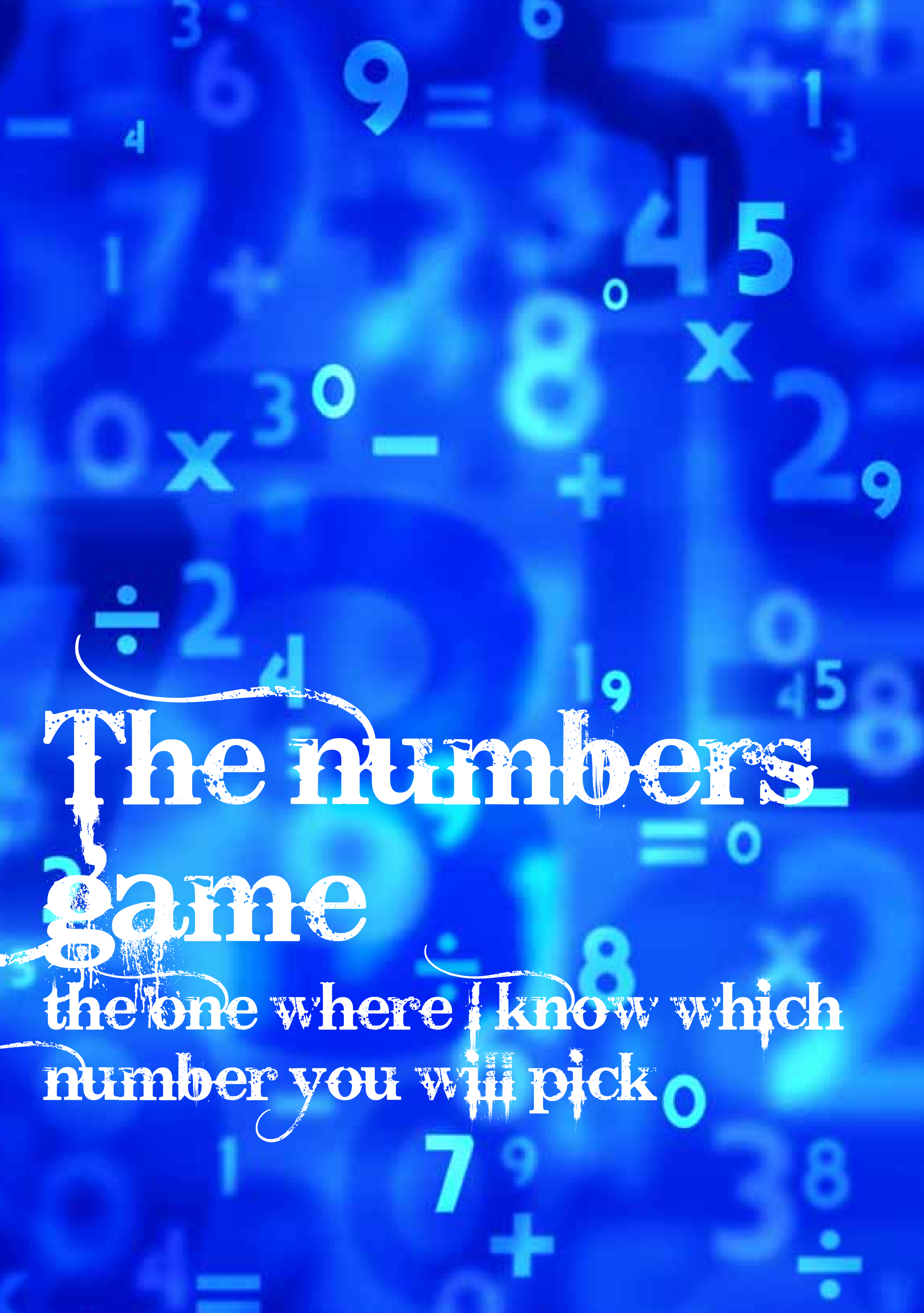
What does this mean to a computer scientist? Well, if you are designing an interface for a task that is hard, sensory cues need to be very, very obvious. Think of an airline cockpit. When the pilot is coming in to land their task is difficult. The procedures they follow (like changing angle of descent, putting the landing gear down, and so on) should not just rely on the pilot's training. Without strong external cues to help in particularly tricky situations, some things may be forgotten. Special care should be taken over steps like remembering the autopilot mode, post-completion steps and initialization steps. Better still, design the difficult steps away if you can.

On January 20th 1992 an A320 airliner crashed while coming in to land at Strasbourg airport. There was a lot going on. The pilot had had to change to a different approach at the last minute. The pilot gave the autopilot a fast speed to descend of 3300 feet/minute. He thought he was entering a shallow angle of descent of 3.3 degrees. He was confused about which mode the autopilot was in. 87 people died; only 6 survived.

## Just a quick one: you said it!

**The one where the tone of voice gives it all away.**

You claim you can tell that a person is lying from the tone of their voice. To prove it you deal the cards into six piles. They chose a card from the centre of one pile and put the piles back together in whatever order they like. They then deal the cards out one at a time, saying: "That's not my card" for each card. When they get to the chosen one you stop them. Their voice gave the lie away. Brilliant bluff detection or just baloney? Find out at **www.cs4fn.org/magic/**



# The numbers game

## the one where I know which number you will pick

# The numbers game: the one where I know which number you will pick

## The magic effect

A spectator calls out 10 numbers. You write them down on separate slips of paper. You shuffle the slips, spreading them out face down and ask someone to pick one. You announce that you know the number they picked...and you are right!

## The mechanics

The person thinks up the first number and you write it down. What your audience does not know is that you then write their first number on every piece of paper! That's the number you announce.

## Prove it works!

All the numbers are the same so of course they pick that number.

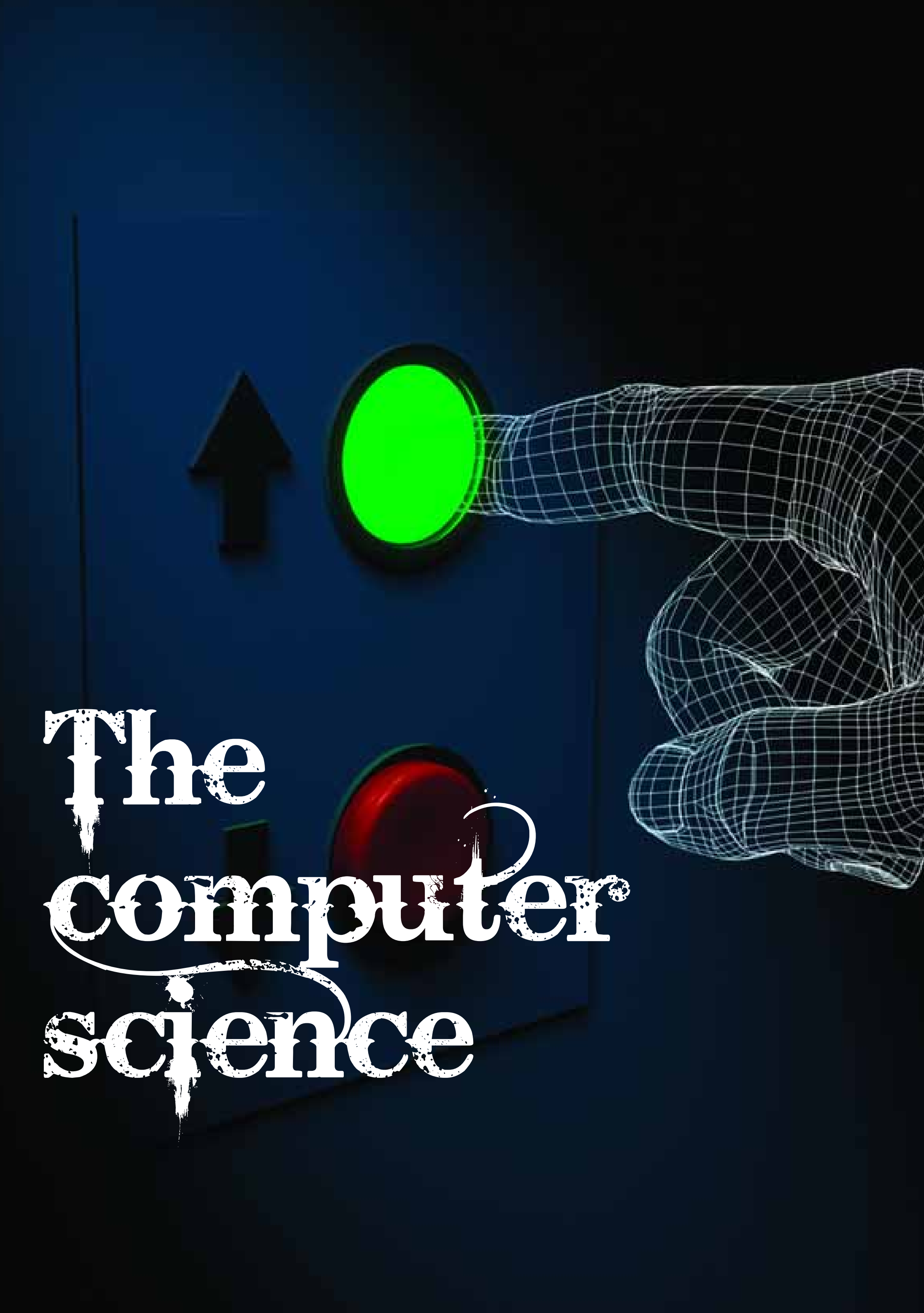
## Showmanship

This works well as a dinner table trick. Try and make it seem really spontaneous. You just happen to have a piece of paper in your pocket and tear it into bits to write the numbers on. That makes it natural to do it by writing numbers down rather than doing it as a 'pick-a-card' card trick.

You can make the trick more powerful. Start by writing down each number the person suggests and let them see what you write. After the first few start to do it in a different way. Do it so that they cannot see that you have switched to writing the same number. Then divide the paper into two piles. One pile is all the same, the other pile a mixture. Ask the volunteer to pick a pile. If they point to the pile that's all the same, you keep that pile and let them choose a single one from it. If they point to the other you say: "OK, you chose to discard those" and still keep the pile where they are all the same. This cunning ploy is often called 'magician's choice'.

## Help pick a card, any card!

Want to take part in some research to help create a new magic trick? All you need to do is surf to **www.cs4fn.org/magic/** and all will be revealed.



# The computer science

# The numbers game: the computer science

## Human-computer interaction

This trick works because what you are actually doing is hidden from the audience. They assume they know what you are writing, but are wrong. They form a **mental model** of what is happening. That mental model is strengthened if they see you writing the first few. They are mentally tracking what is written on the paper, and believe they know what is there because you set up that expectation. The actual state of the system – the values on the paper – are hidden so cannot be used by the audience to correct their faulty belief.

Dodgy mental models can easily lead to errors when using computer systems too. Problems happen if the system is changing its stored values – the state it is in – but these values are not visible on the interface. Then people may start to subconsciously create mental models of how they expect it is working.

Imagine a lift where nothing obvious says what floor you are on. You enter on the top floor and press to go to the lobby. Suppose the doors open before the lobby. It is quite likely you will start to walk out even if the lift had obviously not been descending very long. You were not able to track the hidden state of the current floor so form the incorrect mental model that when the doors open you will be at the lobby.

Mode errors, such as happened in the Strasbourg air crash (see page 35), are more likely to happen if the current mode is not obvious. In that crash, the mode of the autopilot was not obvious. No units were shown on the display with the number. Worse, both 3.3 degrees and 3300 feet/minute were displayed in exactly the same way as 33.

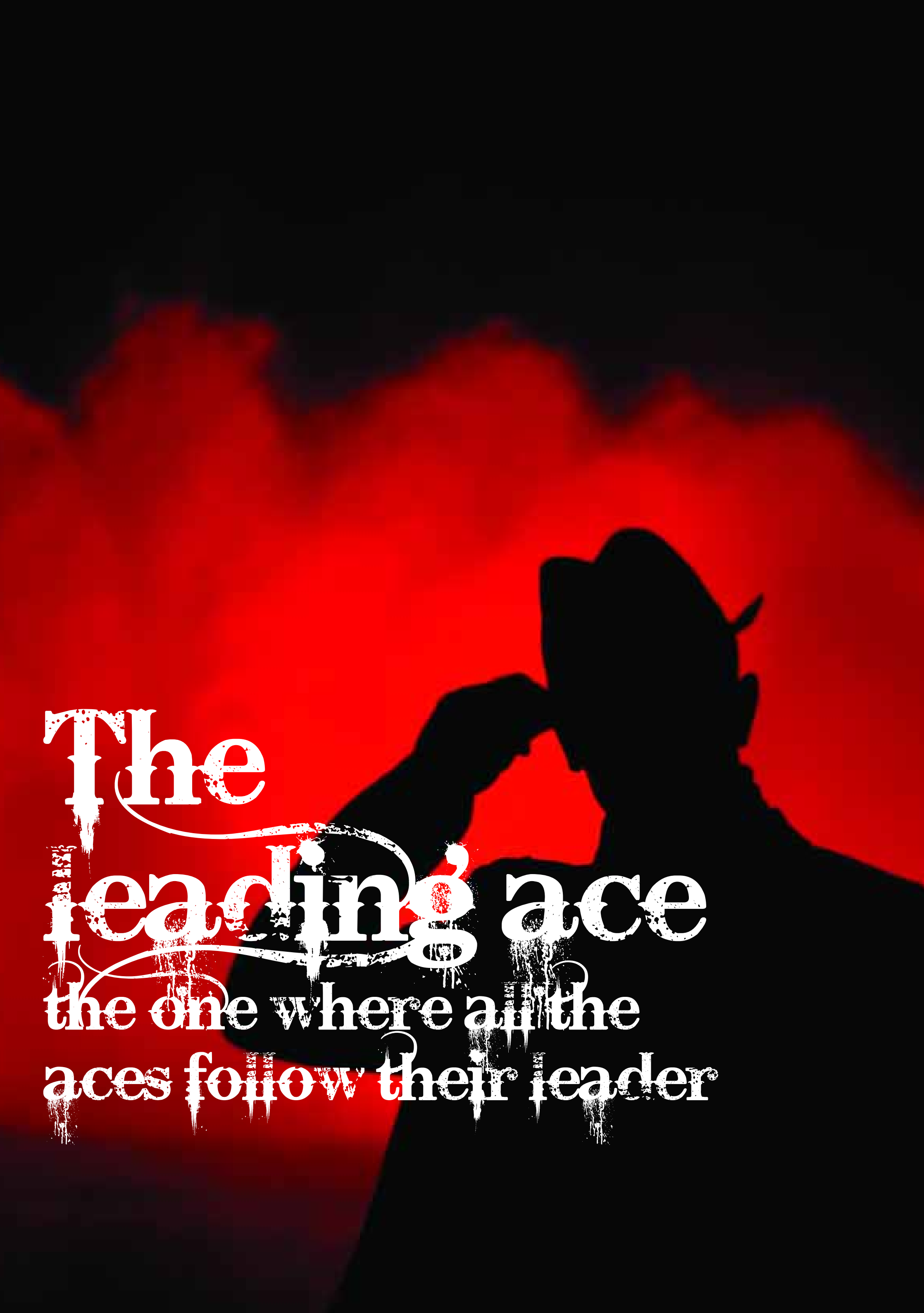
## Design principle

***Make the system state visible.***

It is important to always make critical parts of the internal system state visible on the interface. Use displays, lights and even sounds to ensure the operator knows what state the system is in. That helps prevent a person forming an incorrect mental model. If they do have a wrong mental model they are more likely to realise it if the state is visible.

## Just a quick one: the glass challenge

**The one where a glass is mysteriously tall.** You challenge your friends to a friendly wager. Which is longer: the distance from the table to the top of your glass, or the distance round the rim of the glass? It's a challenge where you can't be right, but in the end you triumph. What does it have to do with an optical illusion and some often forgotten school maths? Find out at **www.cs4fn.org/mathemagic/**



# The leading ace

the one where all the aces follow their leader

# The leading ace: The one where all the aces follow their leader

## The magic effect

The four aces travel from four separate piles of cards joining together in a single pile...except not to the pile where they should have gone!

## The mechanics

First the four aces are removed and placed in a line face up on the table. Three other cards (the 'indifferent' cards) are slowly dealt face down onto each, almost covering them. The pack is put aside. You take each ace in turn, pull it out, turn it face down and place the other three cards on top, squaring them into a pile. You now have four neat, face-down piles comprising an ace at the bottom with three indifferent cards on top. Stack them into a single pile.

Next explain that you are going to deal the cards out. Starting from the top of the pile, deal the first card counting 'one', next to it the second card counting 'two', then deal the third, 'three' keeping the beat of the count going, and finally the fourth card 'ace'.

### Misdirection

Now for some misdirection! Ask the spectator if they have followed the deal so far. Are they sure that the fourth card dealt is an ace? To emphasise the point you use the card on top of your undealt pile as a pointer to gesture to the fourth card and get them to check. Your eyes should look at the fourth card at this point. This is misdirection for the only 'move' in the trick. While the spectator is checking the tabled card, move your hands back and move the card you used as a pointer back, looking like you're going to replace it on top then instead secretly slip it onto the bottom of the deck. Now deal the remaining cards from the pack repeating the one, two, three, ace rhythm.

Because you secretly displaced the top card the three aces are now actually in the third pile.

### Follow the leader

Clear away piles one and two. Turn them over when discarding them to show they are just indifferent cards. This helps 'sell' the previous deal. Everything is happening as the audience expected.

Finally, remove the ace from the bottom of the fourth pile (where everyone expects it to be) and place it face up on its pile. Do the same with the indifferent card from the bottom of the other pile. Now openly swap these two cards. Snap your fingers three times. Announce at each snap that an ace will travel from the ace pile to follow its leader.

Finally, reveal the magical movement.
All the aces invisibly followed the leader.

The aces are not where they should be!

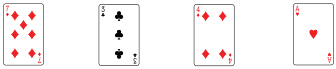
# The leading ace: The one where all the aces follow their leader

## Prove it works!

Each pile originally held four cards: indifferent, indifferent, indifferent, ace. Let's write that as IIIA (using A for ace, I for indifferent: we don't care what's on it).

When we make up the full pile we have IIIA-IIIA-IIIA-IIIA.

Now we start to deal them out:

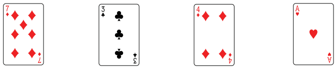


| | | | |
|---|---|---|---|
| I | I | I | A |

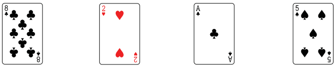
The volunteer can check the fourth card because it is an ace in the place they expect it to be. The remaining cards that are still in the pile are IIIA-IIIA-IIIA.

You use the top card on the undealt pile as the pointer and slip it to the bottom. This leaves the pack as: IIA-IIIA-IIIA-I

Deal out the next four cards and the 4 stacks are now:



| | | | |
|---|---|---|---|
| I | I | I | A |



| | | | |
|---|---|---|---|
| I | I | A | I |

The un-dealt pile is now: IIA-IIIA-I. Do that twice more to give:

| | | | |
|---|---|---|---|
| I | I | I | A |
| I | I | A | I |
| I | I | A | I |
| I | I | A | I |

Your audience think that all the aces are in the last pile, but they are actually in the third one (apart from the one at the bottom). You take it and the one at the bottom of the third pile and place them on top where they can be seen:

| | | | |
|---|---|---|---|
| I | I | A | I |
| I | I | A | I |
| I | I | A | I |
| I | I | I | A |

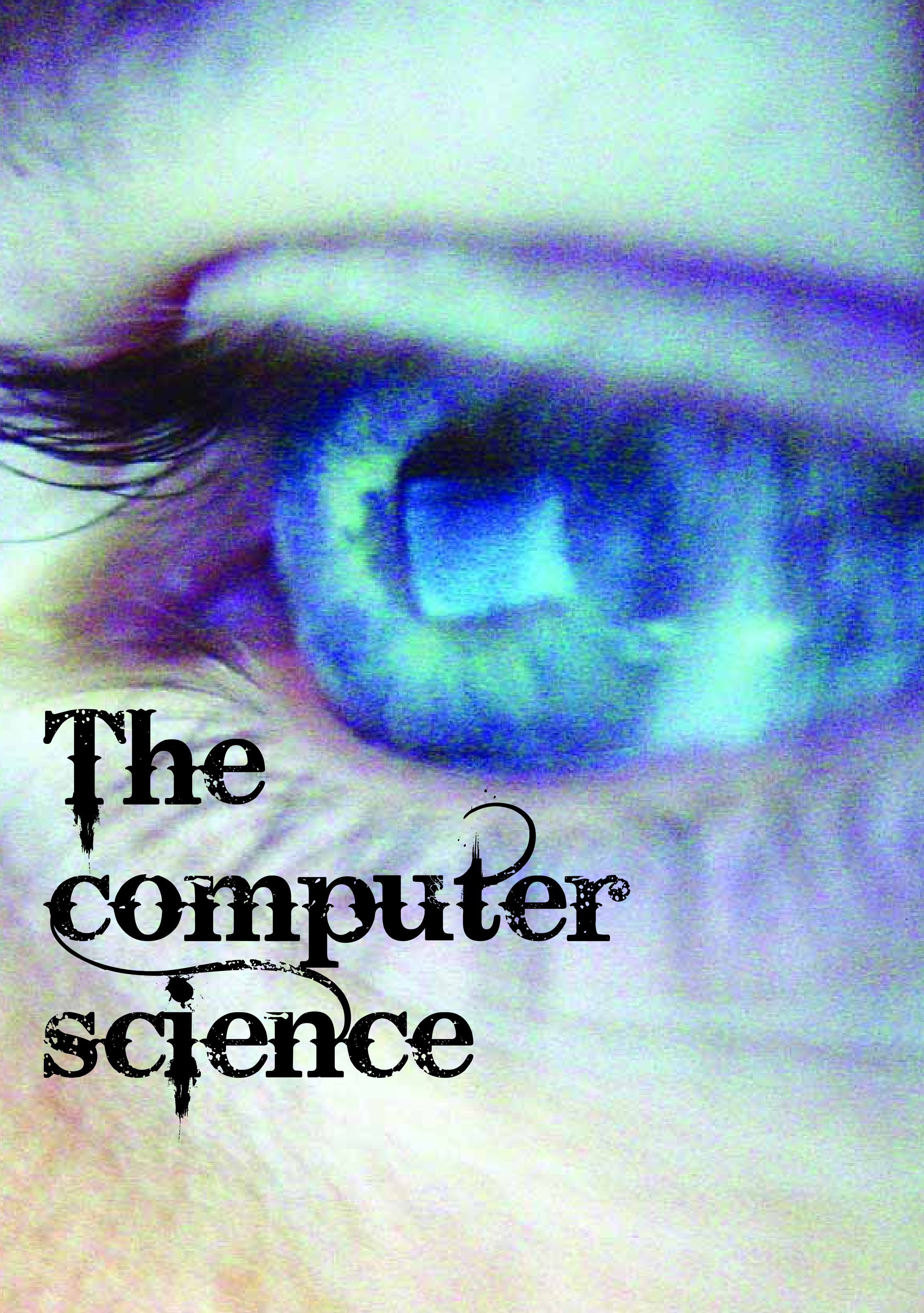
You then take the 'leader' ace that is in the wrong position and swap it to join the others.

| | | | |
|---|---|---|---|
| I | I | A | I |
| I | I | A | I |
| I | I | A | I |
| I | I | A | I |

All the aces are in pile 3 as required.

## Showmanship

In a classic trick from fairground stalls a conman places a ball under one of three cups. He shuffles them round quickly. You watch closely so can say where the ball is, but it is always somewhere else. Build a story about this into your patter but only after the misdirection step so they aren't watching closely then. You do the same with aces instead of balls. To make it easy you will deal them out slowly. Can the punter say where all the aces are?



# The computer science

# The leading ace: the computer science

## The psychology

There are several psychological effects going on here to make the misdirection work. The first is the rhythmic ‘one, two, three, ace’ count. You are conditioning the spectator, and as you prove it’s correct for the first deal, their brain falls into the rhythmic pattern. This is what magicians call a convincer.

Secondly when you are secretly slipping the pointer card to the bottom of the pack you are focusing the spectators’ attention on the card on the table, the ace. People follow social cues. They look where other people are looking. If you look at the tabled card so will your audience.

Finally you are also moving your hands back a little, moving them out of central field, to do the card displacement. Humans have a limited visual area where they attend to what’s happening, and pay less attention to anything outside this area. All the interesting stuff, as far as the spectator is aware, is happening around the tabled card.

Your action of pointing with the card is psychologically invisible. It just looks natural and helpful, not really important, so people tend not to remember it. Once the spectators’ attention has been diverted by the cue, a change to the pointing card can be made.

Researchers have shown that you can make quite big changes to the world and people just don’t notice it if they are paying attention to something else. This is called change blindness, and magicians often use this in their misdirection by making magical gestures, or pointing with their wand to get people to look away from where the real secret move is happening.

## The human-computer interaction

Focus of attention is very important to interface designers. Naive designers often assume that just because they put instructions on the screen that people will read them. They may not even see them! That can even happen if they are really, really salient. If the person’s focus of attention is somewhere else they may have no idea that the crucial message even exists.

The important things have to be placed in positions where a person’s focus of attention will naturally fall. Suppose you are designing an interface where the person has to fill in a form and then, before moving to the next page, hit a save button. If they do not they will lose the information. If the save button is back at the top people may not see it and not do the save.

How do human-computer interaction researchers work out where people’s focus of attention will be? One way is to use what are called **eye trackers,** which work out where someone’s eyes, pointing and track their movements. One way to do this is to just point a video camera at the eye. A computer then processes the image to work out where the gaze is directed.

# The leading ace: the computer science

Eye trackers have shown, for example, that when people look at a web page, they first look in the top-left hand corner. That's why company logos usually sit there (though the Google generation have now learnt not to look at the top)! According to usability guru Jakob Nielsen, the right hand side of a web page may rarely be looked at. People's eyes flick around quickly, often in an F-shaped pattern down the screen. They will scan along the top. If nothing is seen to suck their attention in they will look down the left hand side, scanning across once more lower down. If the person's attention hasn't been captured by then, you have probably lost them. If you are an advertiser you may not want your advert down the right hand side!

## Design principle

***In web page design put the grab top-left.***

When designing web sites, put important things down the left hand side and make sure there is something to grab attention near the top.

## The Persuaders

Much of what a magician does is actually about persuasion. Dominic Furniss, a researcher at University College London, interviewed usability consultants. He found that to be successful, consultants must be good at persuasion not just at usability. They need persuasive powers to win the contract in the first place and to convince the client that the issues found are important enough to be acted upon.

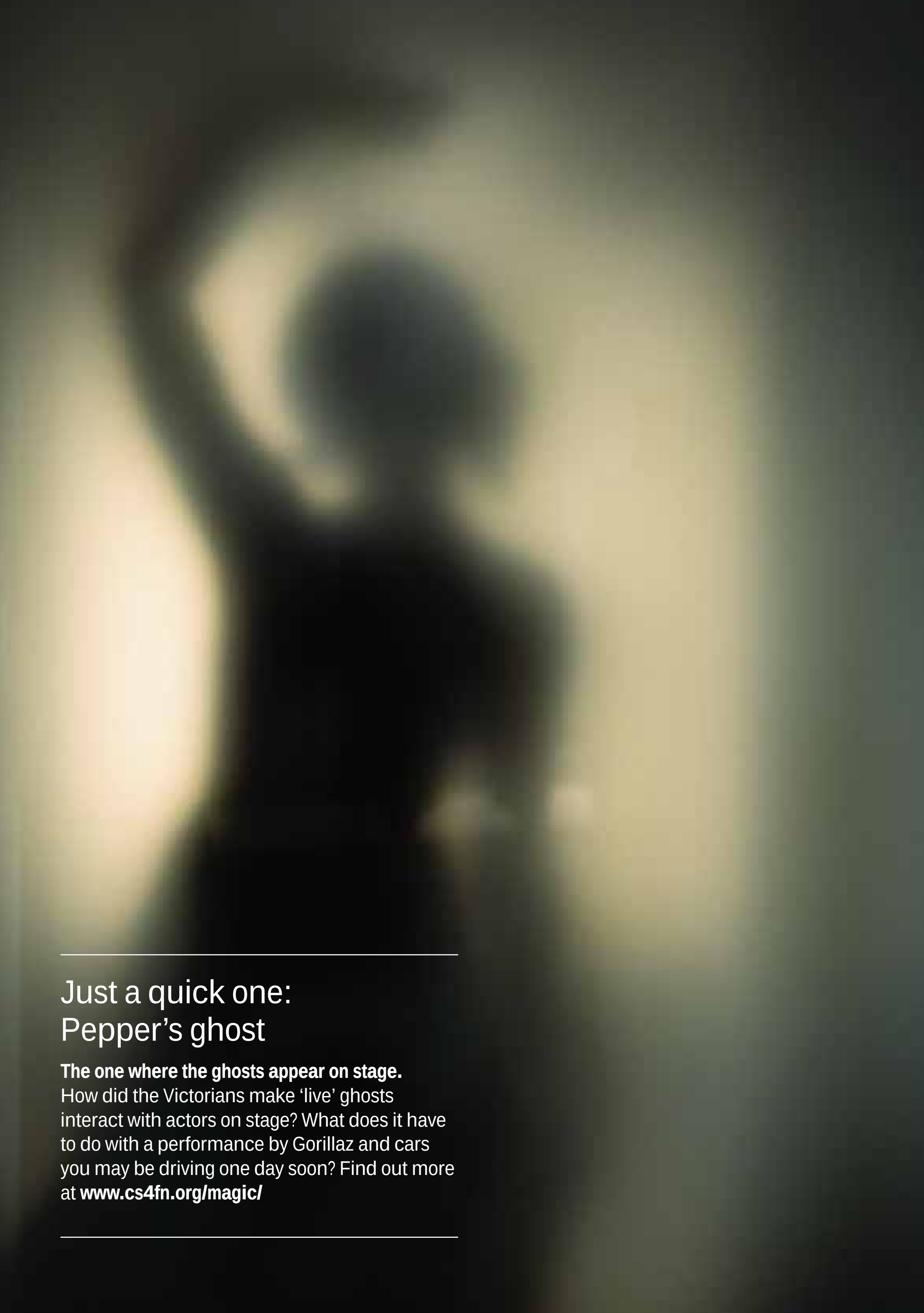
## Do look, don't see!

Vision scientist, magician and psychologist Gustav Kuhn from Durham University has used eye-tracking technology to investigate where the attention of spectators is focused when they watch a magic trick. He found that misdirection is fascinatingly complex.

We aren't always even aware of what our eyes are looking directly at!

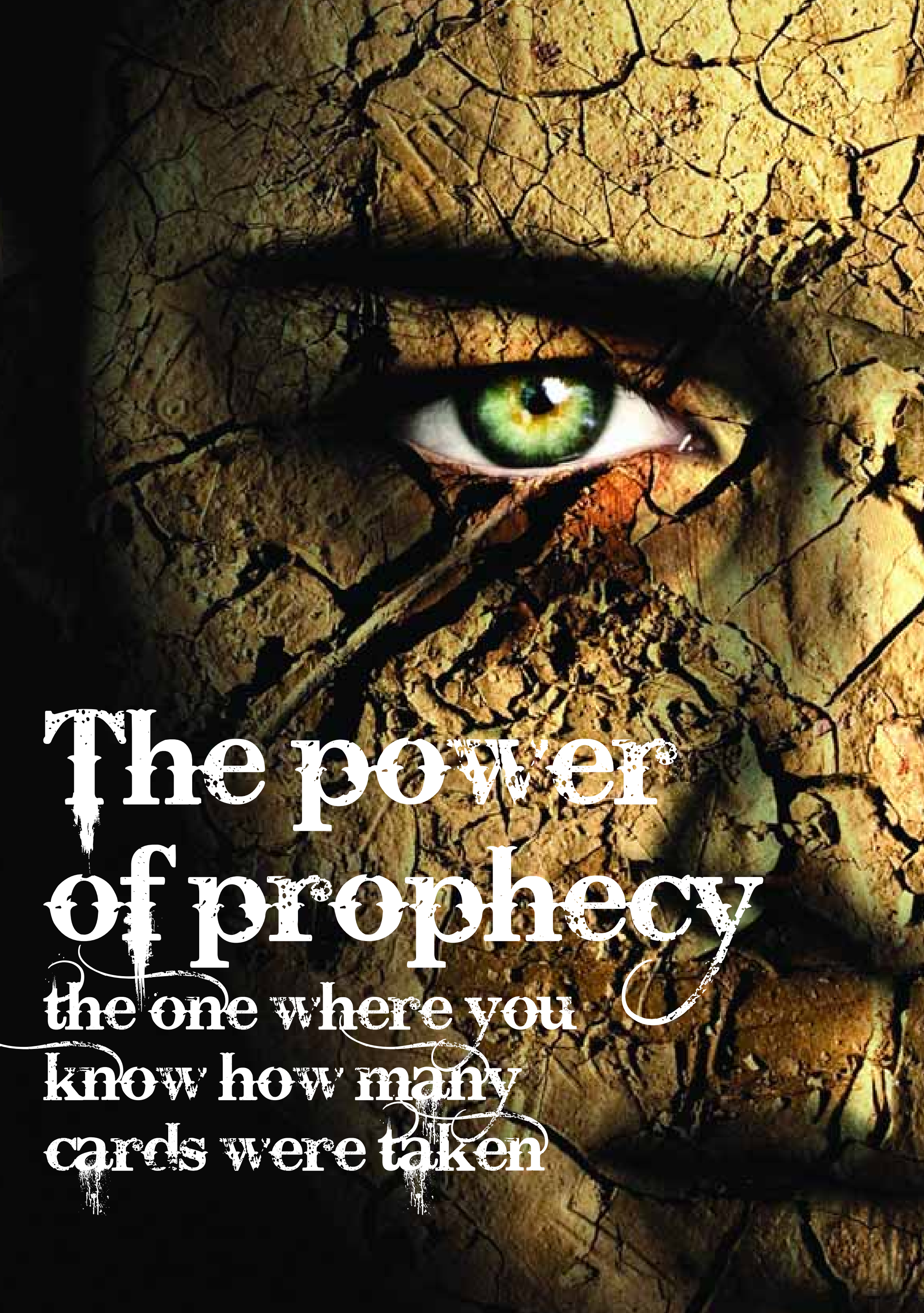
Find out more at **www.cs4fn.org/magic/**

Eye trackers in cars can alert you when you start to fall asleep at the wheel.



## Just a quick one: Pepper’s ghost

**The one where the ghosts appear on stage.** How did the Victorians make ‘live’ ghosts interact with actors on stage? What does it have to do with a performance by Gorillaz and cars you may be driving one day soon? Find out more at **www.cs4fn.org/magic/**



# The power of prophecy

## the one where you know how many cards were taken

# The power of prophecy: the one where you know how many cards were taken

## The magic effect

In this stunning three-phase prediction, you reveal the exact number of cards freely chosen from the pack by the spectator...even though your back was turned when they took their cards.

In magic circles they often say that this is the trick that fooled Einstein!

## The mechanics

Ask a spectator to grab a small number of cards from the top of the pile and hide them while your back is turned. They don't know how many and so you can't know either. You take a bunch of cards too, secretly making sure you take more than they did. You told them to take a small number, so you take a large number.

You now count your cards. You need to work out a 'target' number that is three less than the number of cards you hold. If you have 18 cards, your target is 15. Next pick a suit where you have at least three of those cards but not many more. Turn to your spectator, explaining that you are going to make the prediction in three parts. You make these three statements:

***"I've got as many cards as you..."***

***"I've then got another 3 hearts..."***

***"...and I'm then left with just enough cards to make your number up to 15."***

For each statement you prove what you said was true and move on to the next. So the performance goes as follows:

"I've got as many cards as you..." Ask the spectator to count their cards and place them on the table. Say they have 7 cards. You deal out 7 cards, and put them aside. Part 1 of your prediction is true: you have as many cards as them.

"I've then got another 3 hearts..." You deal 3 more hearts from what's left of your pile. That's part 2 of the prediction correct.

"...and I'm then left with just enough cards to make your number up to 15." In your pile you have 18-7-3 = 8 left. Count your remaining 8 cards onto their 7 cards, and the total is your stated target number 15. Part 3 of your amazing prediction is correct.

# The power of prophecy: the one where you know how many cards were taken

## Prove it works!

The trick here is in the 3-part prediction that looks like you are making three separate but true claims. This step-by-step process builds the suspense and camouflages the simple maths going on. We saw it works if they took seven cards and you took 18. Are we sure it works for any numbers? We could keep checking different combinations (for ever!) or we can do a bit of algebra and prove once and for all it's sure to work.

Say you choose *x* cards and the spectator chooses *y* cards. The *x* and *y* can stand for any number actually chosen.

The first step – "same as you" – means you deal *y* cards down from your pile to match their number, *y*, cards (whatever *y* is). You originally had *x* cards. You are left with the difference. That means you have $x$-$y$ cards left. This step only works if you have more cards than them (i.e., $x > y$). Otherwise you run out of cards.

The second step – "another 3 hearts" – means you deal 3 more cards down. Subtract 3 from the $x$-$y$ left from the first step and you have $x$-$y$-3 cards remaining in your hand (so actually you need $x > y+3$ to have enough cards).

Now for the final step – "just enough to make your number up to…". You did a secret calculation to work out the target number in this step. You counted your cards to get the number *x* (whatever *x* is) and subtracted 3 from it to get the target announced. So it is just $x$-3.

You add your remaining cards to theirs. Will it add to the target number $x$-3 you have predicted? We worked out above that you have $x$-$y$-3 cards in your hand and they have *y* cards. Add them together to give a total of $(x$-$y$-$3)+y$ cards. The *y*s cancel out leaving $x$-3 as you predicted.

Algebra wins again, hidden in your prediction. It's really quite simple and will always be true.

## Showmanship

You can do this trick with other objects too. Ideally you want to be able to say something specific in the middle step though, like "I have three more broken matches", or "I have three more 10 pence pieces". That makes the way the prediction is split seem more natural, and precise.

# The computer science

# The power of prophecy: the computer science

To prove that our magic trick worked, we first created a mathematical model of the system. In this case the model used is very simple – it just consists of the variables, X and Y, to represent how many cards are in different places. What does it mean to say: "the trick worked"? Essentially, it just means that after the steps are done, a desired property holds of the model. Here the desired property is that the cards left in our hand match the target number predicted.

As programs are similar to self-working tricks, we can prove that a program works in the same way. We create a mathematical model of it. We then write a specification of what the model should do – the property to check. Finally, we prove that the specification does hold of the model.

In fact all our 'Prove it works' sections have been doing this for our tricks, though we have used informal reasoning rather than mathematical notation. Logic and algebra have the advantage of being more precise so we are less likely to make a mistake due to fuzzy thinking.

Our model of the system doesn't just have to describe what the program does. We can think of the system as being wider than that including other devices and physical objects like credit cards or keys. We can even model the behaviour of people. At first that sounds silly. People don't just follow rules do they? Surely they behave in all sorts of different ways, so how can we model them and why would we want to?

People behave in systematic ways that can modelled.

Well, they do behave in different ways, but as we have seen there are some kinds of behaviour that are systematic. Even if you can't say what any one person will do on any specific occasion, you can make predictions about general behaviour. You can say that some kinds of errors will be made frequently, for example. If we model the systematic behaviour then we can reason about the consequences of such highly plausible behaviour. If people will often walk away when they have achieved their goal, what is the consequence for a cashpoint design, for example? Are there any situations when that opens a security vulnerability? Would a change of design allow such likely behaviour but without any bad consequences? We can answer these questions with our models.

# The power of prophecy: the computer science

It turns out that we can model this kind of systematic behaviour and it can be used to make predictions about the consequences of plausible behaviour. It gives us a way to verify that there are no design flaws that lead to the kinds of systematic human error that are embodied in the model.

We don't have to do the reasoning by hand. Computer programs called model checkers can exhaustively explore all possible behaviours of the model for us. They mathematically check all possible consequences of the model when interacting with our design. A **model checker** can even tell us the series of steps that would lead to the problem.

## Model-checked psychology

Essentially what we are doing with our model of behaviour is modelling the results of cognitive psychology experiments. We can use the models created to explore our understanding of those experiments. To add the result of a new experiment to a model we first work out a rule that encapsulates the result. We then write it in a way so it can be added to the model. We can then do simulations with it – essentially running experiments on the model instead of on people (much quicker and safer!)

We can do better than that though. We can use a model checker to check all the consequences of our rule and how it interacts with all the other rules in the model. That may show that the rule gives unexpected behaviours we did not see in reality. We can then refine the rule to be more accurate if we know those behaviours aren't possible. Alternatively, we can plan new experiments to explore whether the behaviour ever could happen.

## Messy reality

In psychology experiments it is important to control everything so that there is only one issue being explored at a time. The aim is to find out what the effect of changing that one thing is. In human-computer interaction studies we need to explore what happens when messy reality sticks its nose in, and that is more complicated. We need to understand what happens when lots of things are changing and what the consequences of their interactions are. Model checking can help do that.

Mathematical modelling gives a new way of studying complex cognitive psychology. Mathematical reasoning and empirical experiments work together to give greater understanding of more complex situations.

## Magical snaps from the past

In 1894, before the invention of movies, psychologist and creator of the IQ test Alfred Binet studied French sleight-of-hand artist Raynaly. He wanted to see if the hand was quicker than the eye. He called on chronophotographer Georges Demeny, who captured the tricks with a rapid sequence of still photographs. Only 23 original photographs survive, but when computer animated they show a wonderful performance of the vanishing ball.

# Finale

# Finale

Most people don't do magic – they just watch it for entertainment, never finding out how it actually works.

Magicians start out picking up a magic book, learning a simple trick and trying it on their friends.

Once you have caught the bug you will want to learn more. You read about more tricks and start to learn how to turn a trick into a performance. Now is the time for hard work – lots of practice. It's still fun though. You start to tinker with your tricks and make them better, improving the presentation.

You keep going. You might combine ideas from different tricks so that you have a different effect than either alone. Suddenly you have a new trick altogether. You try it out on an audience and learn from the experience, seeing how it needs to be improved. You now want to learn more of the theory. What is going on in people's heads? How does the mathematics really work? Now, you not only can do the moves others have invented, you understand them and so can put them together to solve new performance problems.

At last you are a real wizard. Now you are one of the people inventing completely new tricks. You are exploring the mathematics and psychology research, looking for new ways to create the same magic-making mistakes. Your tricks are picked up and used by others. You market them and earn money from others using your amazing techniques as well as from your own performances. They even amaze other magicians.

The way to become a software wizard is like that too...

# The computer science

# Finale: the computer science

Most people are just users of technology – they are like the audience. They use the programs the computer scientists write and nothing else. Unlike magic, software is not just for entertainment, it pervades every part of our lives. It's in computer games, the autopilot that keeps a plane flying, and the program that controls a washing machine's cycle.

Some people are interested in more than just playing the games or doing the washing and wonder how to create their own programs. They look at ones other people have written, and read up on how to program. They might create their first program by making changes to an existing one. They've got the bug.

They write more programs using the tricks they see in the programs they come across. They read more and find out about algorithms – ready-made solutions of how to write good programs to solve specific problems. They start to code those algorithms into their own programs and start to modify them to fit their circumstances.

Perhaps they go to university, perhaps they just keep learning as a hobby. If they have an entrepreneurial bent they sell one of their programs. They learn more of the theory, making the most of the work of all the computer scientists who have gone before them.

They are a hot programmer and can create anything they want.

There is a step further to become the sort of person the top companies like Google are looking for. They want special people. People who can do more than just program but who can push the boundaries. They want creative people who come up with new ideas, people who will create the next killer application. They want the wizards.

If you have enjoyed this book then download our first book of tricks from **www.cs4fn.org/magic/**

The Magic of Computer Science

Card Tricks Special

or
A plethora of
pasteboard
paradoxes purporting
the principles of
Computer Science

Presented by
Peter McOwan and Paul Curzon
of the Department of Computer Science,
Queen Mary, University of London
with support from www.cs4fn.org

Queen Mary
University of London

The Magic of Computer Science
www.cs4fn.org/magic/

The discussion of human error research described in this book was inspired by the EPSRC funded HUM project of Rimvydas Ruksenas, George Papatzanis and Paul Curzon of QMUL and Jonathan Back, Dominic Furniss, Simon Li and Ann Blandford of UCL. That on visual perception is based on the EPSRC funded work of Milan Verma and Peter McOwan at QMUL.

This research has been funded by EPSRC grants GR/S67494/01, GR/S67500/01, EP/F02309X/1 and GR/S73723/01.

RESEARCH COUNCILS UK

SELECTED 2009 THE ROYAL SOCIETY SUMMER SCIENCE EXHIBITION

EPSRC
Engineering and Physical Sciences Research Council

**Funded by the Engineering and Physical Sciences Research Council (EPSRC). EPSRC is a member of Research Councils UK, the strategic partnership of the seven UK Research Councils**

**We would like to thank Google for their support.**

Thanks also go to our proofreaders, Ann Blandford, Emily Blandford and Sue White, and to Emma Curzon for showing us the ‘Numbers game’ trick.

**This publication has been produced by the Publications and Web Office for The School of Electronic Engineering and Computer Science, Queen Mary, University of London**

Queen Mary
University of London