The Four Aces

Age group: 7 – adult

Abilities assumed: Nothing

Time: 20 minutes,

Size of group: anything from 1 to 30
Larger groups also possible by using a web cam to project the table top onto a screen

Focus

Computational Thinking: understanding people
Human-computer interaction

Syllabus Links

This activity can be used as a general introduction to from KS2 upwards as to why computing is about more than just understanding technology, as well as the human side of computational thinking. It can also be used as an introduction to the importance of interface design and human-computer interaction.

Summary

You do a magic trick where the audience try to keep track of the Aces in a game of cards. To their surprise the person who they knew had all the Aces turns out to have nothing. It is you the magician who has the perfect hand. How do you steal the Aces with no one noticing? After challenging the audience to work out how it is done, you explain the trick and the link to computing. In doing so you show that computing is about more than just programming and computational thinking is about more than just algorithms. They are about understanding people too.

Technical Terms

Algorithm, computational thinking, interaction design, human-computer interaction, interface.

Materials

Pack of cards
Small table
    Ideally the table should be covered with a tablecloth or similar to make it easier to pick cards up.

What to do

The Grab:
Tell the class you are going to show them a magic trick and teach them how to do it (though if they try it out themselves they should keep the magician's code and not give away the secret – you are allowed to for educational purposes only). Along the way they will learn why computing is about more than just technology. It's about understanding people too.

The Set-up:
Point out that if there are any magicians in the audience who already know how the trick you are about to do works (or bright enough to work it out) they should not shout it out – leave everyone else the chance to puzzle it out. Everyone will have a chance to show how clever they are at the end, if they know how it is done.

Get a volunteer to come to the front. Have everyone else gather round so that they have a good view of the table.

The activity:
Place the four Aces from a pack of cards face up on to the table to make four hands. Explain to the audience this is a trick about why you should never gamble, and certainly not with magicians or computer scientists. Add three normal cards face down on top of each Ace showing the audience the cards as you do so.

Explain you have made four hands that they should assume are the way the cards fell at the end of the last round of the game. In this game whoever has the most Aces wins, so if you can keep track of where the Aces go then you can use that information to decide whether to gamble and so beat the odds. Tell the audience that is what they must do: watch the Aces.

Now turn each Ace face down leaving it at the bottom of its pile. Collect the four piles up and note that if the dealer does not shuffle then you know exactly where the Aces will go. The Aces will be every fourth card. That means, if you deal out four hands, then they will all end up together in the fourth pile. Start to do this to show what you mean. Count as you deal the cards: "1, 2, 3 and an Ace". At this point pause and, pointing to the Ace with the next card from the pile, ask the volunteer to turn it over and show the audience it really is an Ace, "just to check". As they do so, draw the card you pointed with back and slip it to the bottom of the pack, taking the top card in its place. Unknown to the audience, the next Ace is now three cards from the top not four. Tell the volunteer to leave the Ace face up so everyone can see that that is where the Aces are, then continue to deal the other cards out, counting "1, 2, 3, Ace" as you do.

Now point out that one person (suggest it's the volunteer) has all the Aces. They cannot lose. They will therefore gamble high. Now ask the volunteer to point (note "point" not "choose") to two of the other piles. If they point to the first two piles (which do not contain the Aces) then remove both of them and announce that those two players drop out.

If, on the other hand, one of the piles they point to is the third pile (which unknown to the audience has the three Aces) then remove the pile they didn't point to, saying instead that that person drops out. Talk for a while about how the person can't lose so will keep raising the stakes while anyone else stays in. This is important to put a gap between asking the person to point so they don't remember exactly what you did next. Now ask the person to point to one of the two remaining piles. Whichever they point to remove the one that isn't the one with the Aces, saying they finally decide to drop out.

In either case say that you will assume that the hand remaining (the third pile that secretly has the other Aces) is your hand. Point out that, while you remain in the game, the volunteer, who has all the Aces (or so they think!) will keep raising the stakes.

Explain that in this game you are allowed to swap one card with your opponent at this point. The volunteer won't as that would just make their hand worse. You however will. Slide out the face-up Ace from the bottom of the fourth pile. Also slide out the bottom card of the third pile, turning it over to show it is just an ordinary card. Swap those two cards leaving them face up. Remind the audience that though you now have one Ace the other person has three so will still win. Point to the three face down cards in the fourth pile that the audience think are Aces as you do so.

Note that if you were sensible you would have dropped out long ago like everyone else, but you are not going to, instead you are going to try and steal the Aces while no one is watching.

Now tell the audience they must watch the fourth pile really closely. You are going to try and make them all blink at the same time, giving you time to steal the Aces. On the count of three clap you hands in front of the volunteer. Ask them if they blinked. Either way you were quick enough to steal their Aces!

Note that they use to have three Aces and a winning hand but now...turn the fourth pile over and show everyone you've stolen them. All that money gambled has been lost! So who has the perfect hand? Turn the third pile over to reveal you have all the Aces.

Announce that that is why you should never play cards with magicians or computer scientists!

Rather than just explaining how the trick works, first ask anyone who actually saw you steal the Aces to put up their hand (without saying how it was done). One or two may have seen you put the card to the bottom. Then have a show of hands of anyone who thinks they have worked out what must have happened, even if they didn't see it. As you want everyone to be able to do the trick, say you will do it again more slowly. Ask those who have ideas to suggest when they think you stole the Aces: when should people be watching carefully?

Go through the trick again pointing out through the early steps that you haven't cheated yet – the Aces are where they are supposed to be. When you get to the point where you slip the card to the bottom, do it in an exaggerated way. Then, if not everyone saw you do it, take the card back and do it again: point then put it to the bottom even more obviously. The volunteer will find it very hard to watch you and turn over the Ace on the table. (If you have time you could go through the trick again normally before doing it in an obvious way so that some people see and then get them to say when people should watch).

The explanation:
Magic tricks like this involve the magician following a series of steps. As long as they follow the correct steps in the correct order, the magic effect will occur. In this case

the Aces will all end up in the third pile. Magicians' tricks that are guaranteed to work like this, 'self-working tricks'. Computer scientists have another name: an algorithm. An algorithm is just a set of instructions that if followed lead to some effect being guaranteed to happen[1].

This trick has an underlying algorithm, but includes a step that relies on sleight of hand too. It relies on the fact that the audience do not see everything. In sleight of hand tricks, being sure that the mechanics work (i.e., that the algorithm is right) is not enough to guarantee the success of the trick. In particular, the presentation matters too.

In this trick the sleight of hand part relies on simple misdirection. It shows that we have a single focus of attention and we can only focus our attention on a small area at a time. If our attention is drawn to one point then we will miss other things, even if they are there to be seen. Pointing is a very strong social signal so everyone looks at the card. While they are doing so you are free to do things even quite close to where they are looking and they won't see.

What does this have to do with computing?

Magic tricks like this show that it is possible to engineer a system so that everyone makes the same mistake at the same time. Often when people make mistakes it's not about negligence or stupidity but are about the limitations of how our brains are wired. With good design that takes account of those limitations designers can help people avoid making mistakes or recover when they do.

Magicians design magic systems so we do make mistakes, computer scientists must design computer systems to do the opposite. Instead of drawing a person's attention away from things that matter, their attention should be drawn to them using similar 'tricks'. Those tricks rely on an understanding of cognitive psychology: an understanding of people as well as of technology.

Magic trick and programs have the same two important parts:

Magic =	secret method	+ presentation
Program =	algorithm	+ interaction design

It is one thing to write a program to do some task. It is another to make it easy to use. Many programs written are never actually used for real because the people they are intended for find them unusable. In other cases they are much harder to use than they should be. When people make mistakes, sometimes it can just be irritating and time wasting: losing an essay you spent hours writing because you forgot to save it, for example. In some cases the consequences can be impossible to undo: if you set your alarm clock but didn't realize the volume was turned right down so miss that important exam, for example, the bad thing can't be easily undone. In either case, it would be better if the technology was designed to be easy for people to use in the first place. Why can't the computer regularly save your documents itself, for example? Why can't the alarm clock work out that the volume shouldn't be off and do something more sensible?

Computers are also used in situations where the consequences of mistakes can be really bad: like hospitals, for example. Hospital patients are commonly attached to

[1] See our other card tricks for hour to explore the idea of an algorithm more deeply, here we focus on another point: the presentation.

machines that automatically pump drugs into them: like pain relief or cancer drugs to help cure them. Too much or too little and the drug could do more harm than good. The machines have to be set up by a nurse to deliver the right dose. We want that kind of device to be really, really easy to use. When a nurse types the drug dose in we want their eye drawn back to the screen to check the dose before they hit start. We definitely don't want their eye drawn away from it by a flashing start button for example! The designer has to find a way to do that, just like a magician has to design a way to make the audience look where they want.

Computational thinking is the problem solving skill set that people learn when studying computing. It is actually a very similar set of skills to those a magician uses to invent new tricks. Magicians need computational thinking skills too! To a computer scientist a problem isn't solved just because you have come up with a one-off 'answer' like '42'. Computational thinking involves creating algorithmic solutions to problems: tricks that are guaranteed to work, algorithms that a computer can follow blindly to solve the problems, over and over. But computers are used to solve problems for people. It is people who must use the programs developed, so it is important that when coming up with those solutions we think about people and their limitations. People do make mistakes. We must make the programs easy to use. We must especially make sure mistakes are both hard to make and easy to recover from. Computational thinking involves understanding people too.

Variations and Extensions

Write the algorithm and the presentation

Have everyone try the trick in groups. They should then each try to write out their own crib sheet of the basic steps to follow (i.e., the algorithm) so that they can do the trick again at home. They should also write their own story to tell to go with it (perhaps the card teleports rather than being stolen, perhaps it is about cards wanting to be together …) They should then write their own set of notes for each step about the presentation. Which parts of the presentation are critical: which things should be done subtly, when is saying particular phrases important (like saying "point to two piles" rather than "choose two piles"), when it is important to leave time between the steps, and so on.

Further Reading

The Magic of Computer Science

There are lots more magic tricks with computer science twists available from http://www.cs4fn.org/magic/ including several free magic books.

Links to other activities

The Invisible Palming Trick

Teach a trick where the magician invisibly moves a card between 2 piles. This is a fun way to introduce the idea of an algorithm, showing how algorithms are a series of steps that if followed precisely lead to something (in this case magical) being guaranteed to happen – even if the person (or computer) following the algorithm doesn't know what they are doing.

The Australian Magician's Dream

Do a magic trick where you predict a card chosen that even the person choosing couldn't have known. Challenge the audience to work out how it is done, teach them how to do the trick and then use it to explain algorithms, searching, and logical reasoning.

Live demonstration of this activity

Teaching London Computing give live sessions for teachers demonstrating this and our other activities. See http://teachinglondoncomputing.org/ for details. Videos of some activities are also available or in preparation.

Department for Education

SUPPORTED BY MAYOR OF LONDON

COMPUTING AT SCHOOL
EDUCATE · ENGAGE · ENCOURAGE

chi+med
making medical devices safer

EPSRC
Engineering and Physical Sciences Research Council

This activity sheet was created in collaboration with CHI+MED, an EPSRC funded research project involving University College London, Queen Mary University of London, Swansea University and City University on the interaction design of safer medical devices: http://www.chi-med.ac.uk/