Invisible Palming

Age group: 7 – adult, especially good for younger groups

Abilities assumed: Nothing

Time: 15-20 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
What is an algorithm?
Computational Thinking: algorithmic thinking
Human-computer interaction: mental models and feedback (optional)

Syllabus Links
This activity can be used both as a general introduction to what an algorithm is from KS2 upwards is as well as introducing computational thinking. For younger groups it also gives an intriguing use of odd and even numbers.

Summary
You do a magic trick that involves invisibly moving a card from one pile to another. You show the audience they can do it themselves even when they still don’t know how it works if they follow the steps exactly. After challenging the audience to work out how it is done, you explain the trick and the link to computing. In doing so you introduce the idea of an algorithm in a fun way, showing how algorithms are a series of steps that if followed precisely lead to something being guaranteed to happen – even if the person (or computer) following the algorithm doesn’t know what they are doing.

Technical Terms
Algorithm, computational thinking, algorithmic thinking, human-computer interaction, mental model.

Materials
Pack of cards of any kind
(e.g., snap, happy families, normal pack, etc)
If students are to try the trick you need 15 cards per person.
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. Along the way they will learn something about how computers work (including gadgets like their mobile phone).

The Set-up:
Get a volunteer to come to the front and have everyone else gather round the table so that they can see.

The activity:
Ask the volunteer to put their hands on the table so only the fingers and thumbs are touching the table crab-wise – or as though they are playing the piano. Now explain that you are going to do something magical but for it to work, everyone has to chant the special words with you: “Two cards make a pair”.

Take two cards and while chanting “Two cards make a pair” place them together between two of the fingers of the volunteer. Then take another pair and do the same, placing them between another pair of fingers. Do this repeatedly until you only have one card left, getting everyone to say the chant each time. You will have placed one pair between a finger and thumb but have been left with a final finger and thumb slot for the last card. Place the last card there saying “and one left over”.

Next you take the cards back making two piles in front of the volunteer. You do this a pair at a time in the order you put them out. Take the first pair, saying “Two cards make a pair” starting two separate piles with them. Now take the next pair and place one card neatly on top of each pile, while saying the chant. Repeat this until only the last single card is left. Now take this card and say to the volunteer:

“We have one left over. It could go on either pile. It’s your choice.”

Give them the card and let them neatly place it on top of one of the piles. Now you explain that you are going to do something called “Invisible Palming”. They put the extra card on to a pile and you are going to try and invisibly move it. Roll up your sleeve and place your hand over that pile. Rub the back of your hand to “make the card go invisible”. Lift your palm and show that the card you’ve supposedly lifted is invisible. Move your hand to the other pile. Tap the back of your hand, “to make the card drop”. Announce that the card has in fact moved piles.

To demonstrate the magic has worked, first take the pile where the extra card was placed and count off pairs again into a new neat pile – “Two cards make a pair. Two cards make a pair…” and so on. You find the extra card has disappeared – there are only pairs. So where has it gone. Now take the other pile and do the same again, counting out pairs back into a pile. Amazingly after the pairs have been counted out there is an extra card – the extra card really has moved from one pile to the other!

Thank the volunteer and ask for a round of applause from them. Now pick another volunteer and tell them you think they have magic in their blood. You think they can do the trick too. Put your hands out in the piano position and talk them through all the steps of the trick. To their surprise they find that even though they don’t know how they did it, they also managed to move the card.
The explanation:

Give the audience a chance to suggest how it works, then explain it and the linked computing.

A magician would call this a self-working trick. It is guaranteed to work if you follow the steps exactly. It appears magical because you have confused the audience. They believe when you add the last card to a pile you are adding an extra odd card when you are actually making up the last pair – making an odd pile even. There are 15 cards all together. After dealing out the pairs there are 7 cards in each pile. Wherever you place the extra card, that pile will be made up to 8. When you count out the pairs there will be 4 pairs on that pile – so no ‘extra’ card. The other pile will be left with 7 cards, so 3 pairs with one left over to act as the one you pretend has magically moved. You do not have to do anything to make the card move, as nothing has to move!

An easy way to demonstrate this is to just go through the trick again explaining what is happening. Instead of making piles though lay out the cards out so they can all be seen and counted. Match the pairs as you place the second in the pair. When you add the last card point out you have not added an odd card but made up the last pair. There is already an even number in one pile and an odd number so odd card in the other. Finish off the trick showing nothing needs to move to get the magical effect.

What does this have to do with computing? Well, a self-working trick is just an algorithm. Computer programs are just algorithms too. An algorithm is a series of steps to follow that if followed exactly and in the right order will lead to some guaranteed effect. For a trick that is some magical effect, for a program the effect is just whatever the program is supposed to do. In both cases the instructions have to be precise and cover all eventualities. For example, for this trick, it has to work whatever pile the volunteer picks.

A simplified version of the algorithm is as follows

1) Have a volunteer place their hands on the table as though playing the piano
2) Do the following 7 times
   a. Take 2 cards and place them between two of the volunteer’s fingers
   b. Say “Two cards make a pair”
3) Place a final single card between the remaining finger and thumb.
4) Do the following 7 times
   a. Take a pair from the fingers and place them in two piles
   b. Say “Two cards make a pair”
5) Ask the volunteer to place the last card on a pile of their choice
6) Place your hand over that pile, rubbing the back of it
7) Lift your hand, show the palm then place it over the other pile
8) Tap your hand and remove it
9) Do the following 4 times
   a. Take 2 cards from the first pile and make a new pile
   b. Say “Two cards make a pair”
10) Do the following 3 times
a. Take 2 cards from the second pile and make a new pile
b. Say “Two cards make a pair”

11) Reveal that the extra card is now left in that second pile

Programs are written in a language a computer can follow (programming languages). Magic tricks are written in English so that a person can follow the instructions. In both cases they have to be written precisely enough that someone or some computer blindly following the instructions will get the effect even if they have no idea what they are doing. Computers of course are just machines so can only follow instructions (programs) blindly.

A magician who invents new tricks is creating algorithms. A programmer writing a program is doing the same. They are both using computational thinking skills, in this case algorithmic thinking. Algorithmic thinking is what you do when you think up the series of steps that will have the right effect. You are also doing it when you check all the details to make sure it always works. Algorithmic thinking also involves writing instructions really precisely so that there is no confusion about what to do.

Programmers really are wizards!

Variations and Extensions

The whole class do the trick.
Once you have demonstrated it, get everyone in the class to try out the trick, not just one volunteer.

Write the algorithm
Have everyone try to write out their own crib sheet to follow (i.e., an algorithm) so that they can do the trick again at home. In groups get them to test the instructions, checking that no steps are missed and everything is in the right order. Explain how such a set of instructions is just an algorithm.

Mental Models
Magic is all about presentation too, and that links to human-computer interaction. In this trick the magician is building up a false mental model in the audience – they do not keep track of the actual state of the system (the cards) because of the way it is presented. Because of the chanting about pairs, they end up believing the pile has an odd number of cards by adding an extra one when in fact the extra card makes it even. Poorly designed software can do the same thing, if feedback about the internal state of the computer is poor. When you do the trick explaining how it works you are giving a presentation that does not confuse. Similarly a programmer can make it easy or hard to follow what it is doing. With more advanced students the trick can lead to a discussion of this.

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 intelligent piece of paper

*Take part in a test of intelligence against an intelligent piece of paper!*

This is a good introduction to what an algorithm is and how a computer program is just an algorithm. It can also be used to start a discussion on what it would mean for a computer to be intelligent. It can lead on to an unplugged programming activity.

Swap puzzle (forthcoming)

*Solve a puzzle, coming up with an algorithm that your team can follow faster than anyone else.*

This gives a way to introduce the idea of the solution to a problem being a set of instructions that allow others to ‘solve’ it with no understanding. It also explores how different algorithms can solve the same problem but may not be equally good – some may be faster.

Live demonstration of this activity

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