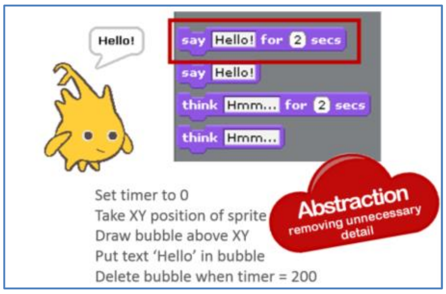
An introduction to the idea of binary being an abstraction for digital representation and an exploration of a variety of ways to introduce and reinforce the key ideas.

Preparation required:
Binary cards, ready folded super switching binary machines and post-it notes for each attendee.
Packs of cards with extra duplicate card and envelope for Australian Magicians Dream.
Demonstration set of binary punched cards, needle/string or skewer for sequel to above.
Binary Bitmap Spreadsheets available in a shared repository. Task sheet for each attendee

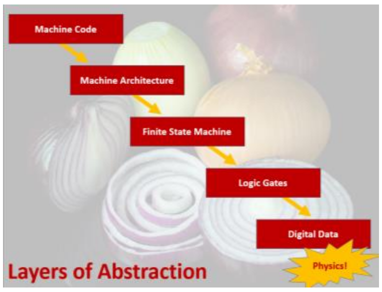
The Digital Advantage

Computational thinking is a particular way of looking at the world. It involves a number of key concepts that occur again and again. CAS have produced a poster that highlights these concepts – great for the classroom wall, so you can keep referring to them whenever the opportunity arises. Consider the idea of abstraction, we use it all the time. Consider, for example a simple function in Scratch. The block hides a lot of mechanical detail – it is an abstraction of the steps required. Abstraction is about the correct level of detail for the problem being solved. In the example our aim is programing the sprite to say things for set periods of time. The 'mechanical detail' is hidden as it's not pertinent to the problem at this level – it is an abstraction of the steps required.



We can think of computer systems as a series of abstractions, rather like our view in Google Earth. To appreciate the world, much detail is hidden. But as we zoom in, views at different levels progressively reveal more detail. The details of what buildings are on which street in Manchester might be important to a taxi driver, but to an airline pilot, a higher level of abstraction might be more appropriate.

Another analogy might be with an onion. You can keep peeling away the layers to expose more underlying detail below. At the level of software, we have a hierarchy of abstraction, starting with a user's original ideas, communicated via an application, the application interacting with the operating system, both probably written in a high level programming language, translated into a low level code that can be run on the specific machine.

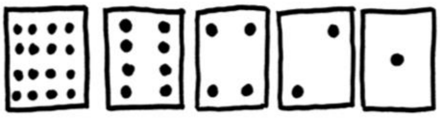


Similarly, at a hardware level we can look at the machine code as an abstraction. The 1's and 0's are bit patterns corresponding to instructions implemented on a particular machine architecture. Those processing operations can be expressed conceptually in terms of a 'finite state machine', implemented through electrical circuits built of logic gates which manipulate the flow of bits. We can go even lower and look at how those bits are actually implemented physically – the flow of electrons, but we're then into the realms of physics.

Not all early computers were digital, but digital representation triumphed because it allowed unambiguous transmission of data over imperfect media. An 'on' or voltage high, might not be quite 5 volts, and an 'off' might not be quite 0 volts, but the difference between the two states meant they could still be clearly recognised. It was the difference, the clarity of distinction, that made the difference. Digital representation triumphed over other types of early data encoding. This series of activities attempts to establish the basics of an understanding of bits, bytes and binary representation.

Establishing The Basics

The link between switches, on and off, and binary is well made in an entertaining clip from Bill Nye, The Science Guy: a long running American TV series in the 1990's. A great introduction for pupils, the clip comes from Episode 18, Season 4, Apl 25 1997. At the time of writing, 31 of the 100 episode run are available on iTunes. Hopefully, the Computer episode, though dated, will be made available shortly. A short 4 minute video from CS Unplugged (00Binary_Counting) is included for teachers unfamiliar with binary numbers.



A quick starter to introduce binary place value requires everyone to have a set of binary cards. You may get a variety of responses to the question, "What do you notice about these cards?" all possibly valid. The point is draw out from the audience whether they can see any patterns. An online version is available to use if you prefer: http://goo.gl/g1Tsl5. The activity establishes that we can make different numbers by adding the values on the cards. A link can then be made between the 1's and 0's representing switches (On and Off).

It takes time to reinforce these ideas, so distribute to pairs of pupils the Super Switching Binary Machine. Children first insert the place value cards, as a reminder. Use Post-It notes to reinforce the link with on/off. Using the presentation prompts as a class activity we can quickly assess understanding. The last 3 introduce pupils to the notion of doing simple addition / subtraction in their head, but representing the answer in binary. The final challenge, calculating the highest possible number using fingers, should stimulate students to work out the binary place values for further columns. Have children look at the pattern again of the place value cards in the machine if need be. Don't tell them the values – the aim is to recognise the pattern.

Once children understand place values beyond those in the immediate activity we can establish the basic units of measurement. When bytes are grouped together, why don't they follow the normal decimal system convention of a 'kilo' representing 1000? The answer lies with binary – our two state system, rather than the 10 state decimal system. Lift the flap on the binary switching machine where the column value cards sit. The values are written as powers of 2. Even if they aren't familiar with the maths, they should be able to see a pattern, and so also be able to predict 2^0 (which does cause some difficulty). Even if the pupils don't grasp the maths, point out that by recognising the patterns they are thinking computationally. Children may be familiar with terms like megabytes and gigabytes, so encourage a little research into the series. Wikipedia will explain why, strictly speaking, we should be using terms like kibibyte and mebibyte, which can lead to an interesting discussion about many ambiguous terms now in use in technology.

There are other kinaesthetic ways to introduce binary. Vi Hart's binary hand dance is a wonderful 3 minute video that provides fun reinforcement: https://youtu.be/OCYZTg3jahU. Perhaps pupils could make videos of their dances, combining a computer science activity with the sort of creative project we'd probably associate with IT?

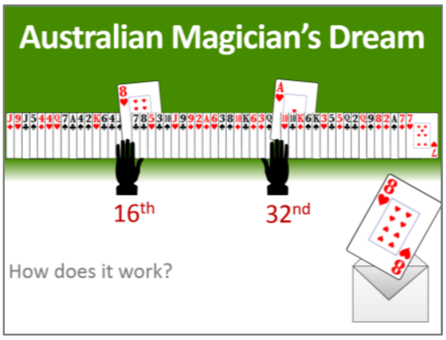
If you wish to take this further, the Wikipedia entry on Finger Binary has supporting images and considers using both hands, as well as representing fractions. See https://goo.gl/ja4Fvs.

Another activity might be to use 4 to 6 children sat in a line on chairs, with a sign showing 0. They act as binary values. As the class chant a count, the correct pupils stand up, holding aloft a 1 (the reverse side of the 0 sign). CS Unplugged have videos on their website of similar counting activities: http://goo.gl/0akYIW.

Finally, Rick Garlikov has an online article about teaching binary using the Socrative method at http://goo.gl/GL7JNF which is also worth reading for an interesting exemplar of how to teach by asking, rather than telling.

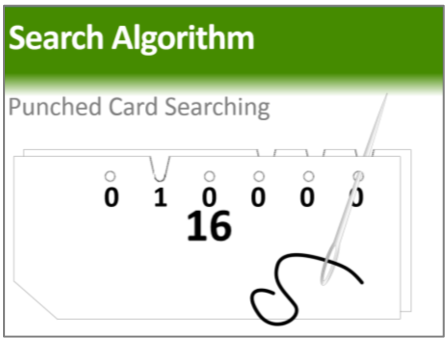
A Little Binary Magic

Paul Curzon and Peter McOwan have produced 3 volumes of wonderful magic tricks to help illuminate CT concepts. Well worth becoming familiar with some of them to expand your teaching repertoire. One great trick, produced as a booklet under the auspices of Teaching London Computing is 'The Australians Magicians Dream'. This outlines a very effective magic trick, gives pointers to the 'patter' and presentation and goes on to explain the links to binary and CT concepts. The booklet is included in the resources.



A sequence of slides demonstrate the trick, with an explanation of how it works. Demonstrate the trick and encourage attendees to then have a go. A deck of cards is spread on a table. Under the table is a sealed envelope. A spectator is invited to split the deck, being asked to choose somewhere near the middle, between the magicians hands. The lower half are discarded. The top half gathered, placed face down, then dealt into two piles 'Down' (face down) and 'Under' (face up) from the top. The 'Down' pile is discarded. Repeat the process with the 'Under' pile until 1 card remains face up. Reveal the same card in the envelope!

How does it work? The deck is prepared in advance, with the 8 of Hearts (the duplicate card in the envelope) in position 16 and a recognised card in position 32. Because the participant splits the deck below card 32, at most there will be 31 cards being dealt out. An animated sequence runs through the cards discarded each time. It will always work, provided the number of cards selected is over 16, but less than 32. We can express the trick as an algorithm. The rules are expressed using just two of the constructs, sequence and repetition, from which all algorithms can be built.



The trick is actually a search algorithm. It was used to search for a particular value in the days when data was stored on punched cards. We can make our own simplified version of punched cards. The template is in the resources. Although it takes some time to prepare, the effect is worth it! There are templates for cards up to the value 63, though you will only use up to 31 to demonstrate the similarity with the magic trick. Each card has holes and slots to represent the binary equivalent of the number on the card. A slot represents a binary 1, a hole a zero; the example of 7 is shown. Like the magic trick, the cards should be kept in order initially. If we pass a needle and thread (or skewer) through the rightmost column, what will happen? All the odd number cards will fall to the floor. These all have slots in that column. It is the equivalent of our 'Down-Under' dealing. We have discarded all the odd numbered cards. We can then continue, passing the needle through the second hole of the remaining cards, thus removing 2, 6, 10, 14, 18, 22, 26 and 30. The procedure can be repeated for each column until only the 16 remains.

Any number can be searched for in this fashion. The search will work with cards shuffled into any order. If we wished to search for seven, binary 000111, pass the needle through each column in turn starting from the rightmost one. But this time retrieve the cards that fall and use those for the next pass. Whether you use the retained or retrieved cards is given by the binary value of the number. So 7 will use the retrieved cards for each of the next two passes, then the retained cards for successive passes until it is the only one remaining.

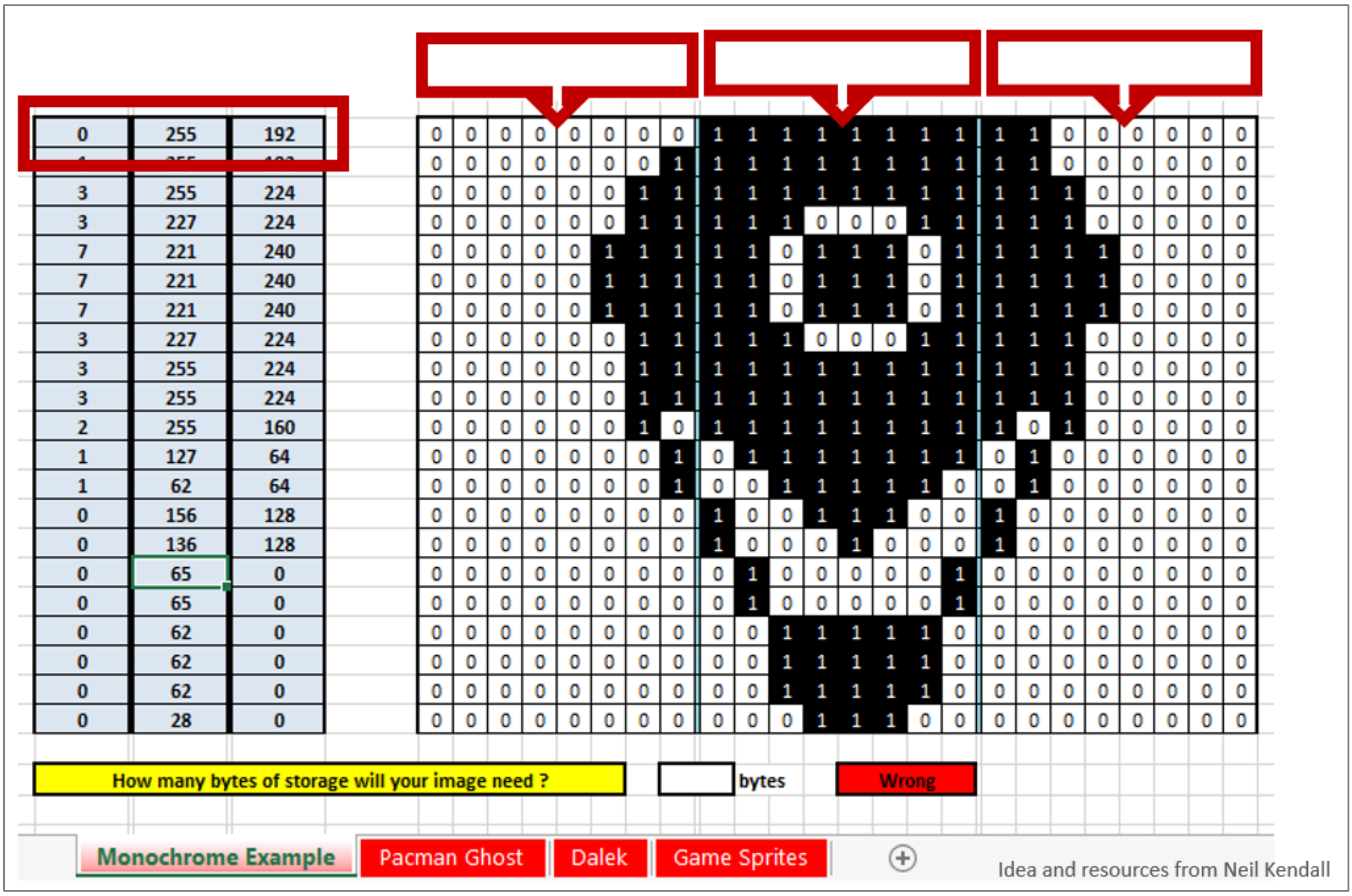
Finally a couple of games worth introducing. Cisco's Binary Game is fast paced, for practising binary / decimal conversion. Children love it! See http://goo.gl/AI2AtX. The 4096 game is slightly less obvious, but is very popular! It helps reinforce the sequence of binary place values. Your goal is to combine the tiles until you reach the number 4096. A task that can be much harder than it seems. It is available online (goo.gl/BlFTRp) or via iTunes, along with other variants. If you feel the need for more reinforcement of basic conversion take a look at the Binary Octopus video: youtu.be/4p-9-nK-mwY.

Bytes, Binary and Bitmaps

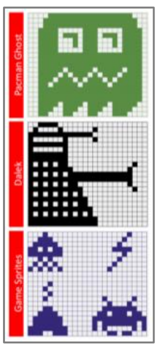
Another popular CAS resource, shared by Neil Kendall allows the teacher to introduce the idea of binary numbers being interpreted as black or white pixel values. Keith Wyles produced a simpler version (shown on opening slide) which may be more suitable with students to start with. Both are included in the resources. We use the harder examples in the presentation.

The challenge is to render simple pictures. It allows pupils to visualise decimal to binary conversions, and gives the opportunity to introduce bytes as a measurement of file size. This activity needs careful explanation. With the correct spreadsheet open, ensure everyone has the correct tab selected, Monochrome Example, so they are looking at the image displayed below. Ensure they read the yellow explanation at the top and highlight the top row of the blue box (as shown).

Take each of the numeric values (0, 255, 192) in turn, and highlight how the binary equivalent represents 8 monochrome pixels in the top row of the image. Pay particular attention to the final one, to ensure pupils see the link between 192 and 11000000.



Finally, point out the challenge to work out the number of bytes required for the whole image.



If further explanation is needed, move to the Pacman Ghost tab. Here the top row is already coloured black as a result of the three 255 values entered. By altering these numbers and observing the result the workings of the spreadsheet should become clear. Suggest obvious numbers to experiment with, such as 1, 3, 7 and 15 so the result is easy to interpret. The task sheet gives the three images to construct (shown left), though choosing one will probably be a long enough activity. It is provided in the resources, three to a sheet so it can be photocopied and guillotined. Students will need a copy in front of them to complete the task. Suggest they write the binary place values above the columns of the chosen image to make calculation of each number easier. This is shown on the final slide, which has each image enlarged.