



Computers: A History
Professor Richard Harvey FBCS

9 March 2021

Several years ago, we decided that we ought to teach Computer Science undergraduates at UEA something of the history and philosophy of computing. It was seen as a rather radical and outré idea – history is for historians not for computer scientists -- we are too busy making the future to bother with old war stories about paper tapes and drum storage. The reason we thought it was valuable was because there is a shared language and culture which comes from the past. Andrew Motion argued similarly about the Bible in 2007 – he described the Bible as an “essential piece of cultural luggage” which infuriated folks who thought the Bible was a bit more than luggage and it infuriated folks who thought the Bible was little more than baggage. Thankfully, computer history is less fought over than the Bible but, like Motion, I do see it as an essential bit of luggage. Thankfully I have not had to, as a professional historian would, weigh the competing claims of various clans to be the first electronic computer (or first mechanical or electro-mechanical computer). Phew. And with that excuse, I can avoid the canonical computer history lecture which is to lay-out the linear history of computers or as Alan Bennett humorously described it as “One ‘effing thing after another”. Instead, I’d like to look at themes.

My cunning plan for this lecture, was to review the literature on what computers actually do in 2021 and use that analysis to construct themes. For example, sometime ago someone told me that the Discrete Fourier Transform was the most commonly executed algorithm in the world, probably because it is used extensively in almost all digital audio and video systems. But, as far as I can tell Ladies and Gentlemen, no such analysis exists. I’m still slightly puzzled that there is no analysis of what computers do and I’ll run the risk of being called a poor scholar if it means you can send me some solid papers on the statistics of computer use. So, in the absence of fact, I’ve segmented computer activity as follows: media (by which I mean the playing of music, video and so on); commerce; games; science and there will be an unavoidable detour into telephony. Now I can hear people in the audience crying “What about?” Well exactly – over to you for the history of the Internet of Things or the history of Computer Vision, AI (although there are some excellent Gresham lectures by, among other people, me, on those things). My general plea is that there is a general need to develop “The history of ...” in IT so if you are attending this lecture, and you know something of the history of IT, or even better have some photos or videos, then please don’t keep them to yourself, post it on a blog somewhere and historians will pick it up and weave it into the tapestry of IT.

Before looking at the themes, I should mention that a special feature of innovation in IT has been the role of industry labs. The rewards for basic research are so great, and so speedy, in our industry that all major IT companies have run Corporate R&D labs. Although there have been a variety of models, unlike Pharmaceutical labs it is common practice for researchers at these labs to be, relatively, free to publish. Some of the significant corporate research labs in IT are listed in Table

11. I have not included national labs in this list: CSIRO in Australia; Qinetiq in the UK, ARPA and Lawrence Livermore in the US and many others also provide an ecosystem targeted most on warfare, intelligence gathering and other strategic issues.

Table 1: Some corporate research labs in IT

| Owner | Location | Size (employees) | Foundation date |
|--------------------|---------------------------|------------------|-----------------|
| Apple | Worldwide | > 1000? | 1980s |
| Amazon Science | Worldwide (25 labs) | > 1000? | 2000? |
| Google | Worldwide | > 1000? | 1998 |
| Nokia Bell labs | Worldwide (founded in NY) | > 1000? | 1925 |
| Facebook Research | Worldwide | ? | 2015? |
| HP labs | Palo Alto & Bristol | > 500? | 1966 |
| Microsoft Research | Worldwide | 1000+? | 1991 |
| IBM | Worldwide | ~ 3000 | 1945 |
| Oracle Labs | Worldwide | ~ 200 | ? |
| Xerox PARC | Initially Palo Alto | 150 | 1970 |

Some of these figures, if true, are quite staggering and the research capacities of several corporate labs now exceed those of nations. For example, the UK is well known to have very high research intensity – there are around 800 Computer Science Professors in the UK – fewer than in several research labs. This has had a very significant effect on the innovation pipeline – many of the innovations we are discussing today were covered by intellectual property cloaks such as national or commercial confidentiality or patents. Naturally, this makes the life of a Computer Science Professor challenging – were I an historian then I would not be expected to compete with teams of 1000+ people paid two or three times what I earn! -- but the commercial labs do provide our industry with innovation at a scale that it is envied in other sectors.

The history of scientific computing is the longest. Indeed, in the early years of computing it was expected to be the *only* use of computers. One of the interesting aspects of scientific computing is how much of the theory was in advance of the technology. Before stored program electronic computers were invented there were scientific methods for numerical integration (Simpson's rule), linear algebra (Gram-Schmidt orthogonalisation), generating Bernoulli numbers (that was Ada Lovelace's work), numerical differential equation algorithms (Runge-Kutta), computer algebra (Hermann), statistical algorithms (such as the Monte-Carlo method sketched by Comte de Buffon), and information theory (Claude Shannon). All these things were devised before electronic stored-program computers and certainly before Computer Science. And it is scientific computing that caused us to think about what we mean by a computer.

Calculating machines have a long history – the British Science Museum has a nice collection, some of which are online, but most people think the history of IT began with a computer that had four properties. Firstly, it had to be speedy. This means that interesting machines like the Zuse Z1, Z2 or Z3 which were electromechanical are ruled out. Secondly, it had to be digital which rules out a group of interesting machines known as analogue computers. Thirdly, it had to be programmable. In the fiercely-fought contest between the UK and US over who developed the first computer this swings the advantage towards the US. However, fourthly, it had to be programmable via software rather than moving switches or wires around. This last adjectival phrase swings the advantage back to the UK.

¹ The table is incomplete because some organisations are coy about how many people they employ. Microsoft, as one might expect, is very open. Facebook, for example, is secretive although I can see their website currently has 439 open positions in research out of a total of 3500 open positions.

Hence digital electronic stored-program computers. I'm sorry that this all a bit nerdy – it's a bit like the famous quiz question, which you should never ask at a pub quiz – “Who invented the first steam locomotive?”² The correct answer has evolved over the years as various factions, adjudicated by historians, made their claims and counter-claims. It depends what you mean by “steam” and “locomotive.” My reasons for focussing on one technology – electronic digital stored-program computers is that that is the dominant technology today and we are looking for echoes from the past. For the avoidance of doubt, I'm not going to use that mouthful for the rest of the talk – I'm just going to use the word “computer”.

The first digital electronic stored-program computer was the Manchester baby. It was a fascinating machine and appears to have bolted together from war spares assembled at tremendous speed, via what is now QinetiQ Malvern, by Tom Kilburn and colleagues. It was reported in short note in *Nature* on the 3rd August 1948. The machine had a memory capacity of 32 words of 31 bits and it was reported that the machine had carried a long division of the number $(2^{30}-1)/31$, had computed the highest common factor of the numbers 314,159,265 and 271,828,183 and had found factors of 2^{18} by exhaustive search (ie trying each number less than 2^{18} in turn). As is so often the case, the Press was considerably ahead of reality and *before* the first computer, the press had contained exaggerated claims of what computers could do. And this was because of ENIAC (the Electronic Numerical Integrator and Calculator) which was devised in the USA. ENIAC was probably operational as early as Dec 10th 1945 and it was fast and it was programmable but, programming it involved moving switches and wires, and it was really designed to do one thing only which was to compute artillery firing tables. Nevertheless, it was well publicised and before long, certainly as early as 1946 the Press in the UK and US was awash with Giant Electronic Brain stories. In the UK most of those stories were started by Lord Mountbatten who gave a rather hagiographic annual speech to the Institute of Radio Engineers³ which got picked-up by *The Times* and *The Daily Mail* in which he indulged in speculation about how the ENIAC might soon be cooking dinner. Meanwhile the impressive engineering and crypto-science associated with much of the war effort was secret⁴. A special mention must also be made of CSIRAC which was only months behind the US and UK machines but, more importantly, was completely preserved.

Leaving aside the first uses of computers, which were usually military, it is the scientific sphere which was associated with early computing developments – it was an early adopter of high-level programming languages with FORTRAN, see previous lectures. Nowadays barely any science takes place without computers and even number theorists find themselves proving theorems in PARI (a computer algebra system). Early computers were very constrained in their speed and precision so led to the development of a new field known as numerical analysis which is the design and analysis of algorithms that will execute speedily and reliably on real computers⁵. Later scientific use of computers tended to focus on simulation of complex systems. One well-known use of computer power is Finite Element Analysis in which a physical system is broken up into small enough chunks

² I've learnt over the years that so long as you don't answer Robert Stevenson you can gain a spurious reputation for being an intellectual. If you are in Scotland, then I should say William Murdoch (1784). In the US one should defend John Fitch (1794) and anywhere else one might hazard Richard Trevithick.

³ These dinners continue to this day – “The IET dinner” is a notorious event in our calendars. Each year around February one wonders if some wealthy engineering firm will be kind enough to invite you to this event. I suspect that Lord Mountbatten, who was speaking at the Savoy, was given a less rowdy reception than some recent speakers.

⁴ Most authors now agree that ENIAC was scooped to the title of first electronic computer by Colossus which was the valve computer designed to decode encrypted messages sent via a cypher known as the Lorentz cypher. The first Colossus was operational in 1943. But this is contested territory – there are advocates for the Atanasoff-Berry computer and the Zuse Z3 as being the first electronic digital computer.

⁵ There is an entertaining talk by Nick Trefethen of the Oxford Mathematical Institute which attempts to show that many such algorithms were developed by Mathematicians who were also Professors and were supported by the government [14]. From my perspective it is hardly a surprise to learn that algorithms that are frequently used by mathematicians were developed by mathematicians nor is a surprise to learn that Mathematicians are keen to show governments that they are a worthy investment!

such that each chunk can be modelled by the type of equation we learn to solve as undergraduates (a plate or a beam for example). These solutions are analytic, but they are then coupled together in a jumbo system of numerical equations known as boundary conditions. The computer solves these huge equations and hence the whole system is modelled. This Finite Element Method (FEM) is now the standard technique in structural analysis, mechanics, fluids, and similar fields and is notorious for using CPU cycles. More recently the “killer apps” for scientific computing have been in bio-computing. The current response to the covid pandemic is highly directed by scientific computing which is essentially matching strings and classifying them as similar to known strings to “sequence” a sample of biological material⁶. The most recent innovation has been the application of machine learning to scientific problems. In my experience, scientists can be wary of machine learning, firstly for the human reason that no-one wants to be replaced by a machine, and secondly because some machine learning algorithms, give an answer but no insight into how it was obtained. However, when machine learning can model previously unmodellable problems, such as protein folding, then some of the reticence may be overcome⁷ [1].

Notwithstanding the early developments in scientific computing, the business people were quick to realise that many problems would be automatable. In previous lectures on Data [2] I mentioned Sabre [3] which is one of the more venerable business systems – a system for booking plane tickets. Sabre was an example of something that is familiar to us now: an IT system that completely transformed the industry. However, the early business computers, by which I mean digital electronic stored-program computers, were variants of the early scientific machines and were not expected to transform or define an industry – usually the goal was to save labour or time or reduce errors in tedious administrative procedures. Binac, which morphed into UNIVAC, derived from ENIAC. Leo derived from EDSAC which was an early research computer at the University of Cambridge.

I have mentioned UNIVAC in previous lectures in connection with Grace Hopper [4], the inventor of the first compiler. Hopper also designed a business programming language called FLOW-MATIC that became COBOL. FLOW-MATIC was designed to appear to be not too mathematical (mathematics was thought to be off-putting for some reason) and COBOL inherited the English-looking syntax. A COBOL program is split into sections called divisions and may use the GO TO command which is unusual in a modern programming language. Needless to say, it is widely reviled by computer scientists due to its verbosity and alien structure⁸.

Leo was a fascinating story [5] and arose from the observation that Lyons teashops (the Café Nero of their day) could usefully be automated. Lyons were already world-leading in their ordering and stock-control systems and had sent two senior managers to the USA to study business methods. They met with the ENIAC team and returned to the UK with ideas for computerising their business. They then made some seedcorn investment in EDSAC, the first electronic computer built in Cambridge and by 1951 they had their own computer, the Lyons Electronic Office or LEO I which was used for inventory, payroll and valuation calculations. Given the market-leading performance of the LEO range of computers, I must admit I was surprised to learn that there is some doubt if the LEO computers actually provided any operational or financial benefit for Lyons but from our perspective LEO was important not only because it was the first computer, or set of computers, to tackle the standard batch-jobs of business, but it also introduced the concept of bureau computing (LEO ran Ford UK’s payroll for a while) which remains important to this day (we call it Cloud Computing now).

⁶ It is fair to say that without fast string-matching algorithms there would be no genetic sequencing of Covid.

⁷ The protein folding problem is as follows. Given a sequence of amino acids then we ought to have enough information to work out the shape of the protein in 3D (how it “folds”).

⁸ I had intended to include some COBOL code so we share our mutual revulsion, but it is so verbose that space has run out. The IBM Knowledge Centre has plenty of examples if you fancy experiencing the horrors of COBOL.

So far there has been no sign in my history of what most of us experience in commercial software: office utilities for typing; arithmetic and speech-making. The history of word processing, which has now been studied from a literary perspective [6], probably starts with the concept. As with quite a few computer inventions, the concept has a longer history than the actually practical device. One theory is that it, or rather *textverarbeitung*, was invented by Ulrich Steinhilper a who was an IBM salesman⁹. IBM made a typewriter which also made digital recordings of the characters that could be uploaded to other systems. This “Mag Card II” typewriter cost around \$11,000 in 1976 but could not really be called a word processor. Early word processors were specialist computers made by companies such as Wang and Lanier. They aimed to bridge the gap between linotype machines, which were used to typeset high-volume books and newspapers, and typewriters. But it wasn’t until the early 1980s that programs that ran on general purpose computers came to prominence. The three popular programs were Wordstar, Wordperfect and Word. Wordperfect was the dominant program for five years or so but two things led to the dominance of Word: the emergence of Microsoft’s windows operating system (Wordperfect had been developed for the text-based DOS and it was too slow to transition) and “bundling”¹⁰. Word won and it now dominates the word processing market. Interestingly Word has been steadily losing market share as freeware competitors such as Google docs and Apple Pages which are doing to Word what Word did to WordPerfect – offering an inferior product for free in combination with other services¹¹. Alongside these programs for the home-user market, we ought to mention mark-up languages which are computer languages which instruct printers and other display devices to “mark-up” the page. Modern printers are often not sent bitmaps describing the page, for the very good reason that the resolution of a modern printer is several thousands of dots per cm and we don’t want our wifi networks crawling to halt everytime we print a page. Instead, we send them a postscript program¹² which the printer renders onto the page. But there are also mark-up languages for typesetters (roff was the Unix typesetting language which is an ancestor of RUNOFF which was developed in the mid 1960s). Roff, or it’s later variant, Troff has been largely replaced by TeX or its later variant LaTeX. My impression is that almost all serious papers in Mathematics and Computer Science are prepared in Latex¹³ and despite it not being a layout editor, it remains extremely widely adopted due to its portability, lack of bugs and lack of “lock-in”¹⁴.

The ubiquitous spreadsheet has a more complex history because no-one is very confident what constitutes a spreadsheet. The LANPAR system of 1970 is often mentioned because it was covered by a patent¹⁵ that included the concept of forward and backward referencing. In subsequent spreadsheets such as SuperCalc and Multiplan, the user had to repeatedly hit the refresh button. LANPAR avoided this by keeping tabs on the references. Notwithstanding the ingenuity of LANPAR, the first spreadsheet program that resembled a modern one was Visicalc as it gave users a WYSIWYG¹⁶ display, automatic recalculation, range copying and formula building. Visicalc was a vital boost for the Apple II computer as it was only later ported to other computers such as the Atari

⁹ Somewhat astonishingly, he was also a WWII Fighter Ace who having been shot down over England devoted considerable energies to escaping from his Canadian internment camp.

¹⁰ In the 1980s Microsoft was notorious for aggressive pricing policies that essentially put competitors out of business. By including Word for free with other Microsoft products, the consumer was faced with some free software, Word, that was almost as good as the pricey WordPerfect.

¹¹ In the aggressive world of IT it is truly the case that those who live by the sword often die by it.

¹² Postscript is a language developed by team including Utah alum John Warnock who founded Adobe. PDF is just a compressed form of Postscript.

¹³ I have a friend who is a very distinguished Computer Scientist who confided to me that he felt confident rejecting manuscripts not prepared in LaTeX since it was a clear indication that the authors didn’t know much about the field.

¹⁴ Lock-in is a “feature” that is designed to stop you moving from one system to another and was a commercial tactic widely deployed in the eighties and nineties to prevent users switching technology.

¹⁵ It turned out to be a heavily contested patent which was finally resolved 25 years later with the US Court of Appeals calling the patent “unenforceable” but the LANPAR patent story had a great number of twists including the patent attorney switching sides!

¹⁶ WYSIWYG = What You See Is What You Get

and Commodore machines. Some commentators have called it the first “killer app”¹⁷. Excel was late arrival, 1985, and was not very popular until the same two factors that came into play with Word were extant: as Microsoft transitioned from DOS to Windows, some of the competitors, Lotus 1-2-3, were slow to port their systems, and secondly bundling – although the early versions of Excel were nothing special, they were free as part of a bundle of software from Microsoft.

Before PowerPoint, business presentations used either black boards or flip-charts (which were cheap), overhead projectors (a loud whirly thing that allowed one to write on acetate slides with special pens) or, if one had the budget, 35mm slides¹⁸. Although early data projectors, or data panels were very expensive and dim, the flexibility of presentation software has made the PowerPoint presentation ubiquitous leading to the familiar complaint of “death by PowerPoint”¹⁹. In July 1987, Microsoft bought its first company, Forethought, and PowerPoint was brought to widespread attention. PowerPoint did have predecessors, in the video industry one could spend hundreds of thousands of dollars on machines that could produce frames for news backdrops and so on and there were a few alternatives dating from the early eighties that ran on home computers (HP-Draw and Slidemaster) but, unlike Word and Excel which were essentially inferior copies of previous programs, there is some agreement that PowerPoint was something new and made available novel and useful features.

We have now dealt with the two dominant computing applications for early computers on what were called *mainframe* computers. The nub of the issue with mainframe computers was that they were eye-wateringly expensive. My own department replaced its own computing machinery in 1974 at a cost of £343k (in today’s prices roughly £3.9M)²⁰. Clearly these were major investments so it was very difficult to obtain computer time for what might be perceived as dilettante interests such as computer media. Nevertheless, in the dark hours, when computers were not crunching the payroll or performing weather forecasting, there were groups of programmers investigating some of the things that we now use computers for.

Digital media, by which I mean the playing of music, video and games has an interesting history. Digitised audio and video took a while to develop because the interfaces (analogue-to-digital converters (ADCs) and the inverse, digital-to-analogue converters (DACs)) took some time to develop and they were very expensive, but it is relatively easy to get a squeak out of a computer – you connect an amplifier and loudspeaker across one of the data or address lines, and then you write a program to alternately switch the line high or low. That creates a square wave. It sounds rough, because that’s what square waves sound like (there is a Gresham lecture that discusses this [7]) but, by varying the length of time before switching the line, one can vary the frequency. Experts believe that the first computer to play music was the Australian computer CSIRAC probably in 1950. No recordings were made, but there are reconstructions. The oldest extant recording was made by the BBC in 1951 of the Ferranti Mark 1, which was a commercial version of the Manchester Mark 1. Both of these systems were fun, but they suffered from two faults: firstly, that the program had to reconstruct the music in real-time which was possible for square waves, but far too demanding if one wanted to reconstruct, say, a violin. The solution was to add additional analogue circuitry such as oscillators and noise generators which themselves could be controlled by computers. A key

¹⁷ One definition of a “killer app” is a program so desirable that you are prepared to buy a new computer to run it.

¹⁸ So ubiquitous has PowerPoint become, that there is now a Swiss political party that campaigns against its use [15].

¹⁹ As far as I can tell, the phrase “death by PowerPoint” is not a complaint about the ubiquity of it, nor of the lengths of PowerPoint talks; it is a complaint about the over-use of hackneyed symbols such as bullet-points, handshakes and gear-wheels.

²⁰ I should note that the idea that any UK or Australian university would spend £4M on departmental computing nowadays is eccentric. Computing departments in UK universities are cash cows, so most Vice-Chancellors aim to spend as little as possible on Computing in the hope that their surpluses can support other less viable subjects. The USA and Asia have less marketized Higher Education systems, which is one explanation for the success of technology-based subjects in those countries.

development was MIDI (the Musical Instrument Digital Interface) which allowed interoperability between different programs and instruments. MIDI arose from a co-operation between the leading electronic instrument manufacturers [8] and was responsible for the boom in home composing as it allowed people without musical training to create notated music and sounds. But what of the role of computers in recorded music and video? To see that, we need to digress into telecommunications.

The telecommunications industry saw the benefit of digital signalling from the beginning – indeed the very earliest forms of electrical communication were digital. The principal challenge of early telegraphs was that long-distance cables had poorly controlled capacitance and inductances which meant that, even if microphones, loudspeakers and linear amplifiers had been invented, they would have had very restricted range [9]. Morse-code or other digital signalling codes were much easier to send long-distance²¹. Gradually the limitations of human operators were overcome and some important developments were codes which required less typing (the Baudot code) and tape readers and teleprinters which meant telegraphy could run at hundreds of words per minute. All this was established before digital computers.

The system of converting real-world signals (analogue) signals into digital codes for transmission and conversion back to the real-world is usually called Pulse Code Modulation or PCM. To my surprise there are patents referring to PCM as early as 1921. The Patent, due to Paul M Rainey of Western Electric describes a five-bit fax machine. Certainly, long before stored program electronic computers, a set of theoretical tools and practical devices had been established for PCM. The famous Bell labs in New York state had established a prototype PCM system by 1947. This was quite feasible as, in principle, programmable computers are not required for telecommunication systems, all that is needed is some agreed protocols for how, say speech, is to be encoded; input and output devices and way of switching²². However, it was that network of telephone cables and the switching technology²³ that stalled developments in telecommunications. Although the telecommunications companies had a seemingly unassailable lead in digital media capture, reproduction and transmissions, they squandered it due to their anxiety about upgrade costs to their networks and worries that transmission of voice data via the internet would undermine their monopolistic positions.

History does not record when analogue-to-digital converters were first connected to programmable computers, but it is fair to say that throughout the history of programmable computers there has been the potential to attach interfaces to convert analogue signals into digital words (because the interfaces existed before computers). As is usual with technology development, not only has miniaturisation led to dramatic cost reductions but also the digital domain has been extended closer and closer to the analogue device. As an example, consider audio output. The loudspeaker in your mobile phone is little different from those in use 50 years ago. However, whereas previously the computer would have been outputting 16-bit words (the PCM signal) which would then go into a set of very balanced resistors (a digital to analogue converter) and then into an amplifier which in turn drove the speaker, we now send the digital signal straight to the loudspeaker. The width of the signal sent to the loudspeaker is adjusted digitally using a system called pulse-wide modulation. It sounds fine and, unlike the DAC/amplifier system which might be only 50% efficient, PWM Class-D amplifiers can easily reach over 90% efficiency. Likewise, connecting cameras used to involve some intricate format conversion – video cameras being designed to output TV signals which have a complicated legacy format – but nowadays the camera is digital and appears to the computer as a stream of bits.

²¹ Which is not to say that it was trivial to compensate for the impedance of telegraph cables.

²² Furthermore, when telephone networks use programmable elements, danger lurks, as we can see by Western governments anxieties over Huawei.

²³ AKA the Public Switched Telephone Network or PSTN.

However, for the distribution of digital media to be viable it was essential that there was an efficient distribution mechanism. It is rumoured that a few musical acts released recorded music on floppy diskettes, but I have not been able to validate this²⁴ so digital music was not a practical reality until the invention of the Compact Disc in 1982. A 650Mb disc could hold around 60 minutes of uncompressed stereo audio recorded at a sampling rate of 44,000 samples per second²⁵. If that same disc was used to hold uncompressed video information, then it would hold about 2 seconds! Thus, began an important technological pairing which is, firstly, the provision of high-bandwidth connections to the places of work, homes and ultimately your mobile device and, secondly, compression. I've talked about the principles of data compression in earlier lectures [10] and I hope to revisit the practicalities of the topic next year. It has a fascinating and colourful history with fierce legal battles and some big personalities such as Leonardo Chiariglione (the MPEG man). It was the provision of the internet that really changed music distribution, although the rise and fall of the CD as a music distribution mechanism makes a fascinating interlude.

Also, in the category of media we find computer games. Game playing has been associated with computers since the 1940s when Alan Turing started to write an algorithm for playing chess. Unfortunately, Turing discovered that the Ferranti Mark 1 was not powerful enough to run his algorithm so he had to test it by hand²⁶. When it comes to video games then there are a number of efforts that used custom or semi-custom hardware: Bertie the Brain which was a Canadian computer built by Josef Kates in 1950 to noughts and crosses (tic-tac-toe to Americans); Nimrod which played the game of Nim and was built by the Ferranti Company in 1951. Nimrod was inspired by an earlier machine called the Nimatron built for the 1939 world fair. It is difficult to be precise about what was the first computer game because of the doubt around the definition of a computer. What is certain that computer games did not take-off until two essential components were developed: a high-quality display device which was initially the cathode ray tube and enough processing and memory for the computer to respond in real-time. Probably the first time these all came together in one place was in a machine called a Digital Equipment Corporation (DEC) PDP-1 or possibly its predecessor the TX-0 (pronounced tixoh) which was an experimental computer built at the Massachusetts Institute of Technology (MIT²⁷). The TX-0 and PDP-1 were important devices because they used transistors which meant the system weighed under a tonne and has considerably more reliability than previous devices. It is often said to be the start of the "hacker" culture because it was easy to open up the back and plug-in one's own hardware modules²⁸ and the layout of machine with a keyboard, a display device and a printer all to hand encouraged rapid iteration programming rather than the more systematic approach of previous years. There is therefore wide agreement that Spacewar, which was a game written by a team of graduate students and employees at MIT, was a significant step in computer history. The PDP-1 was a significant machine, possibly the first mini-computer. DEC went on to become one of the most important computer companies in the world before being wrong-footed by microprocessor-based computers and being bought by Compaq, who subsequently were subsequently bought by HP²⁹.

²⁴ Brian Eno did distribute his album Generative Music 1 on floppy disc but that is a program that plays music and it plays differently each time it is run. As we know from previous lectures, the program for doing something ought to be smaller than the final artefact it encodes. There are also a few modern bands for aesthetic and nostalgic reasons have crammed tunes onto floppy discs using modern audio compression techniques.

²⁵ In earlier lectures I discussed sampling but the basic idea is that analogue signals can be perfectly reproduced, without error, so long as they are sampled more than twice as fast as their bandwidth. Human hearing runs to about 20kHz so 44.1kHz is just about adequate for domestic audio.

²⁶ It was reconstructed for the Turing 100 celebrations when Gary Kasparov beat it in 14 moves but on being interviewed afterwards Kasparov commented that the principal deficiency was due to the restricted search capabilities of the machine.

²⁷ Everyone writes MIT to avoid the difficulty of spelling Massachusetts)

²⁸ Valve-based computers required backbones that carried live DC voltages at sometimes hundreds of volts. This made them deadly and a potential deterrent to graduate students who wished to plug-in experimental hardware.

²⁹ One in a series of misguided mergers and acquisitions.

Spacewar brought to the fore the importance of computer graphics in consumer computing. Superior graphics capabilities have been the saviour of many computers, particularly domestic computers over the years³⁰. The modern windows operating systems that are now ubiquitous date from a systems called Sketchpad devised by Ivan Sutherland at MIT in the early 1960s³¹. Sutherland later moved to Utah where he built one of the best Computer Graphics departments of all time. The list of achievements is too long to describe here but the origins of Adobe, Silicon Graphics, Netscape are to be found in Utah alumni³². Subsequent to early developments in rendering and graphics, it became evident that computer graphics was an incredibly computationally intense activity. The root cause of the problem is not that the algorithms are especially complex, it is the vast numbers of pixels that have to be filled-in or “shaded” and the rate at which modern displays have to be updated. This led to development of additional computers which could be attached to the main computer and hence speed-up performance. Nowadays we call these Graphical Processing Units (GPUs). Early models were to be found in Arcade games and were highly customised to particular games. As computer games moved to 3D there was an opportunity for relatively cheap processing to pull-away from domestic computers and hence the games console was born. Their continuing popularity is a bit of puzzle but the competitiveness has recently declined with only three manufacturers now making consoles (Nintendo, Sony and Microsoft). One spin-off of this activity was the availability of cheap GPUs which have proved incredibly useful to scientific computing – the covid sequencing results and advanced machine learning that I referred to in the Science section could not have been computed without very large numbers of Nvidia GPUs.

Bibliography

- [1] A. W. Senior, R. Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, A. W. R. Zidek, A. Zidek, A. W. R. Nelson, A. Bridgland, H. Penedones, S. Peterson, K. Simonyan, S. Crossnan and P. Kohli, "Improved protein structure prediction using potentials from deep learning," *Nature*, vol. 577, pp. 706--710, 2020.
- [2] R. Harvey, "Data: past, present & future," Gresham College, 24 Nov 2020. [Online]. Available: <https://www.gresham.ac.uk/lectures-and-events/data-future>. [Accessed Feb 2021].
- [3] R. V. Head, "Getting Sabre off the Ground," *IEEE Annals of the History of Computing*, vol. 24, no. October-December, pp. 32--39, 2002.
- [4] R. Harvey, "An introduction to programs," 2 Feb 2021. [Online]. Available: <https://www.gresham.ac.uk/lectures-and-events/programs-intro>. [Accessed Feb 2021].
- [5] G. Ferry, *A Computer Called LEO: Lyons tea Shops and the world's first office computer*, London: Fourth Estate, 2003.
- [6] M. G. Kirschenbaum, *Track Changes: A Literary History*, Harvard University Press, 2016.
- [7] R. Harvey, "Speech Processing: How to Wreck a Nice Peach," Gresham College, 27 Nov 2018. [Online]. Available: <https://www.gresham.ac.uk/lectures-and-events/speech-processing>. [Accessed Feb 2021].
- [8] C. Anderton, "Craig Anderton's Brief History of MIDI," MIDI Association, [Online]. Available: <https://www.midi.org/midi-articles/a-brief-history-of-midi>. [Accessed Feb 2021].
- [9] D. de Cogan, "Dr E.O.W. Whitehouse and the 1858 trans-Atlantic Cable," 2009. [Online]. Available: <https://atlantic-cable.com/Books/Whitehouse/DDC/index.htm>. [Accessed Feb 2021].

³⁰ The hardware of the original Macintosh computer had little to recommend it, but it had an integrated windows operating system and mouse which meant it could be used by non-specialists and was therefore a hit.

³¹ Sutherland's supervisor was Claude Shannon – the inventor of Information Theory.

³² Thanks to a colleague at UEA, Robin Forre [3]st for pointing out the significance of Utah.

- [10] R. Harvey, "It from bit: the science of information," Gresham College, 23 October 2018. [Online]. Available: <https://www.gresham.ac.uk/lectures-and-events/it-from-bit-science-of-information>.
- [11] N. J. Higham, "The accuracy of floating point summation," *SIAM Journal on Scientific Computing*, vol. 4, no. 4, pp. 783--799, 1993.
- [12] R. Harvey, "An introduction to Algorithms," 20 October 2020. [Online]. Available: <https://www.gresham.ac.uk/lectures-and-events/algorithms-intro>. [Accessed 17 Nov 2020].
- [13] R. Harvey, "The Digital University and Other Mythical Creatures," 11 Feb 2020. [Online]. Available: <https://www.gresham.ac.uk/lectures-and-events/digital-university>. [Accessed 17 Nov 2020].
- [14] N. Trefethen, "Who invented the great numerical algorithms?," 2005. [Online]. Available: <https://people.maths.ox.ac.uk/trefethen/inventorstalk.pdf>.
- [15] M. Poehm, "The Anti-PowerPoint Party," 2011. [Online]. Available: <http://www.anti-powerpoint-party.com>.

© Professor Harvey, 2021