The Aotearoa Digital Arts Reader
Edited by Stella Brennan and Su Ballard
Designed by Jonty Valentine

ISBN: 978-0-9582789-9-7

A catalogue record for this book is available from The National Library of New Zealand

Title: The Aotearoa Digital Arts Reader
Author/Contributor: Brennan, Stella (ed); Ballard, Su (ed)
Publisher: Aotearoa Digital Arts and Clouds

Aotearoa Digital Arts Trust
www.aotearoadigitalarts.org.nz

Clouds
PO Box 68-187, Newton, Auckland 1145
Aotearoa New Zealand
www.clouds.co.nz

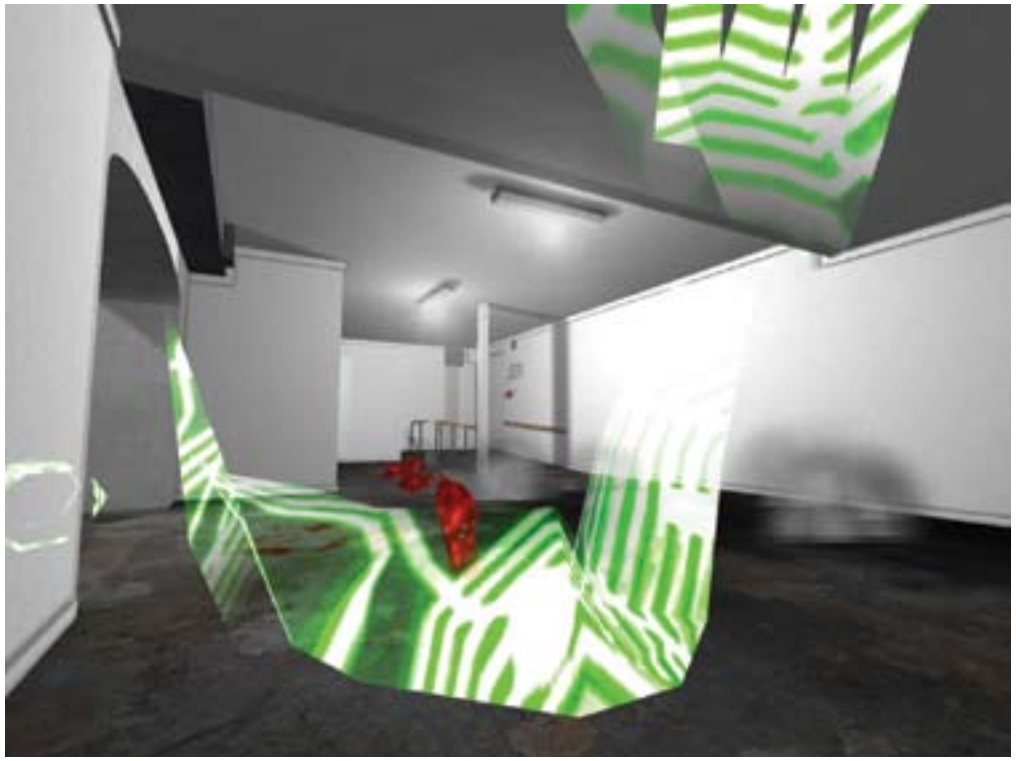
Every effort has been made to trace the copyright holders of the illustrations reproduced in this book. Unfortunately, this has not been possible in all cases. The editors and publisher would be pleased to hear from any copyright holders whom they have been unable to contact and to print due acknowledgement in subsequent editions.

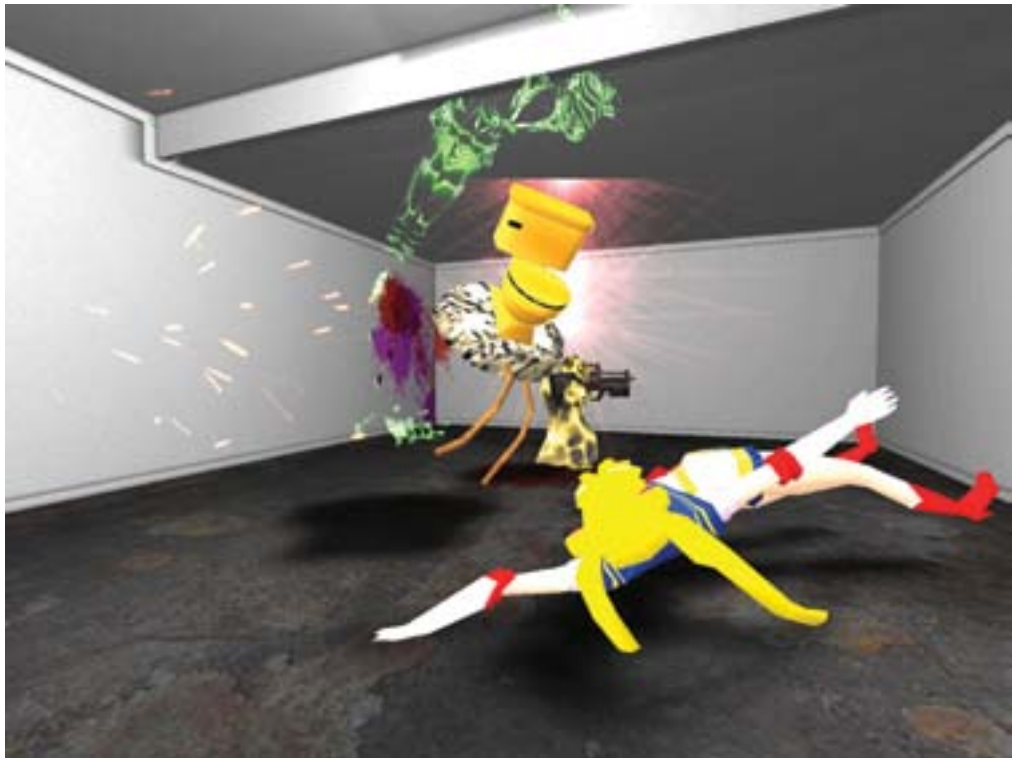
Unless otherwise noted, all images are repoduced courtesy of the artists.

Editing a book takes a long time, and many people have helped along the way. Stella and Su would like firstly to thank the authors and artists who have contributed to this book, and the institutions and individuals who shared their image archives with us. We would also like to acknowledge the work and support of the following: Nova Paul, Leoni Schmidt, Col Fay, Khyla Russell, Justine Camp, Letitia Lam, Pam McKinlay, Geoff Noller, Sarah McMillan, Robert Leonard, Melinda Rackham, Mercedes Vincente, and Gwynneth Porter, Deborah Orum and Warren Olds from Clouds. And of course, Jonty Valentine for the hours spent in design. Thanks also to the ADA community, and especially to the other ADA trustees, Janine Randerson, Douglas Bagnall and Zita Joyce.

Thanks most of all to our families: Nathan, Moss and David.

The *Aotearoa Digital Arts Reader* would not have been realised without the support of AUT University, Otago Polytechnic and Creative New Zealand.





Morgan Oliver, *New Work for PCs*, 2007, real-time projection, game mod.

1980s Home Coding: The Art of Amateur Programming[1]

Melanie Swalwell

Writing code oneself was a key part of the reception and culture of early home computers; systems such as the BBC, the Spectrums, the TRS-80, the Atari, Commodore and Amiga ranges, and the Sega SC3000.[2] In the 1980s, home coding was a significant use of these computers, both in terms of the numbers of people who dabbled at coding, and as a mode of engagement with a then–new technology. A highly experimental practice, it presaged many of the contemporary practices involved in digital culture, the often-discussed phenomena of appropriation, modification, and remixing. Yet while the 'advent' of Web 2.0 has raised the profile of productive consumers, remarkably little attention has been paid to the earlier practices of home coders.

This essay focuses on the *experimental* basis of home coding in the 1980s, drawing on archival and interview-based research into the New Zealand reception of computers and digital games during this decade.[3] This research into home coding enables us to develop a clearer understanding of the uses that people made of home computers. After the French theorist, Michel de Certeau, I suggest that we know very little about what people actually *did* with these early items of digital consumer technology.[4] Though some accounts do exist, these tend to be more concerned with either the spectacular (hacking) or the feared potentials of the 'computer revolution', such as job losses, thus they provide only partial understandings of early engagements with digital technology.[5] One reason why home coding may have been overlooked is its *everydayness*, its homeliness, if you like. Even today, those who dabbled at writing software at home—after school or on the weekend—typically consider that their activities were unremarkable, expressing their sense that 'everyone was doing it.' Unfortunately, this popularity does not guarantee that home coding will be remembered; indeed, many of the creations of this era—dubbed 'hobbyware' by one of my informants—have already been lost.[6]

I have conducted in-depth interviews with people who were active home coders, and in this essay I blend extracts from the accounts of Katharine Neil, Mark Sibly and Simon Armstrong, Fiona Beals, and John Perry, with material from other informants (including the founding editor of *Bits and Bytes* magazine and technology journalist, Neill Birss) and archival sources. I focus here on the twin issues of how my informants learnt to code and what it was that they wrote. In many, if not most cases, the simple answer to the question of what they wrote was, 'games.' I have pursued informants who are knowledgeable about early games, because games were often a key reason why people purchased or otherwise acquired a computer. An important driver not just of the development of early home computers—or 'micro-computers', as computers for the home or office user were then called—games also drove the uptake of many early home computer systems, as Neill Birss observed. Though they are often deemed unworthy of serious consideration, digital games are significant in the histories of both home computer use and amateur coding.

1. This research was made possible by grants from the Faculty of Humanities and Social Sciences, and the University Research Fund, Victoria University of Wellington.

2. This last system attracted immense interest in New Zealand, probably because there was little commercial English-language software available for it, at least to start with; because of this gap, people rose to the challenge and wrote their own. Collectors have compiled a list of literally hundreds of software titles that were locally written. See A. Wheeler and M. Davidson, "SC3K Tape software list," *Sega Paradise*, 28 May 2005. http://homepages.paradise.net.nz/atari/sc3000lists.html. This system is the focus of a pilot project by VUW's NZTronix research team, of which I am a member: we plan to port and re-distribute an early locally written game title for use on a mobile phone platform. For more information, see http://www.nztronix.org.nz

3. New Zealand is a specific case in the reception of early home computers with its own idiosyncracies. Research indicates that the advent and arrival of early computers in New Zealand was uneven *vis à vis* the rest of the world, both in terms of the systems that were brought in and the issue of delay or 'lag.' Nevertheless, a number of similarities exist between the New Zealand reception of home computers and that of other geographic contexts. Some of the research detailed here will, therefore, resonate with the histories of computing in other locales. See Melanie Swalwell, "Early Games Production in New Zealand," paper presented at *Digital Games Research Association Conference,* Vancouver, Canada, 17 June 2005, and Melanie Swalwell and Loyer, "Castoffs from the Golden Age," *Vectors: Journal of Culture and Technology in a Dynamic Vernacular* 3, April 2006. http://www.vectorsjournal.org

As early adopters of digital computers, home coders were effectively inventing uses for home computers. Though a number of magazines advocated the usefulness of programs to help with household budgets, and word processors to compile recipe collections, none of these could be considered the 'killer-app' of home computing. Compared to other technologies whose use is clearly part of, or implied by, their function, early computers were a technology in search of a use. On their own, they were essentially *useless*. Programming was the only use that was indigenous to the computer. As Katharine Neil explains:

> *…you couldn't really do much with computers back then unless you learnt a bit of code. You'd do really dumb, primitive things, but… In those days, people bought games and they'd play games, but the coolest thing was to write stuff yourself. In those days, you bought a computer and you bought a book on how to program it, and there was only one way you could do it! And if you didn't do it, then what was the point of having a computer, because it didn't do anything, it didn't do anything for you.*

Learning to code

In 1983, Colin Boswell predicted that "computer education would take place in two places—the school and the home."[7] While computer classes were on offer in some schools (and the debate about computers in schools was heated and long running), and enrolments in formal computer science degrees rapidly escalated during the 1980s, many programmers were self-taught, at least initially.

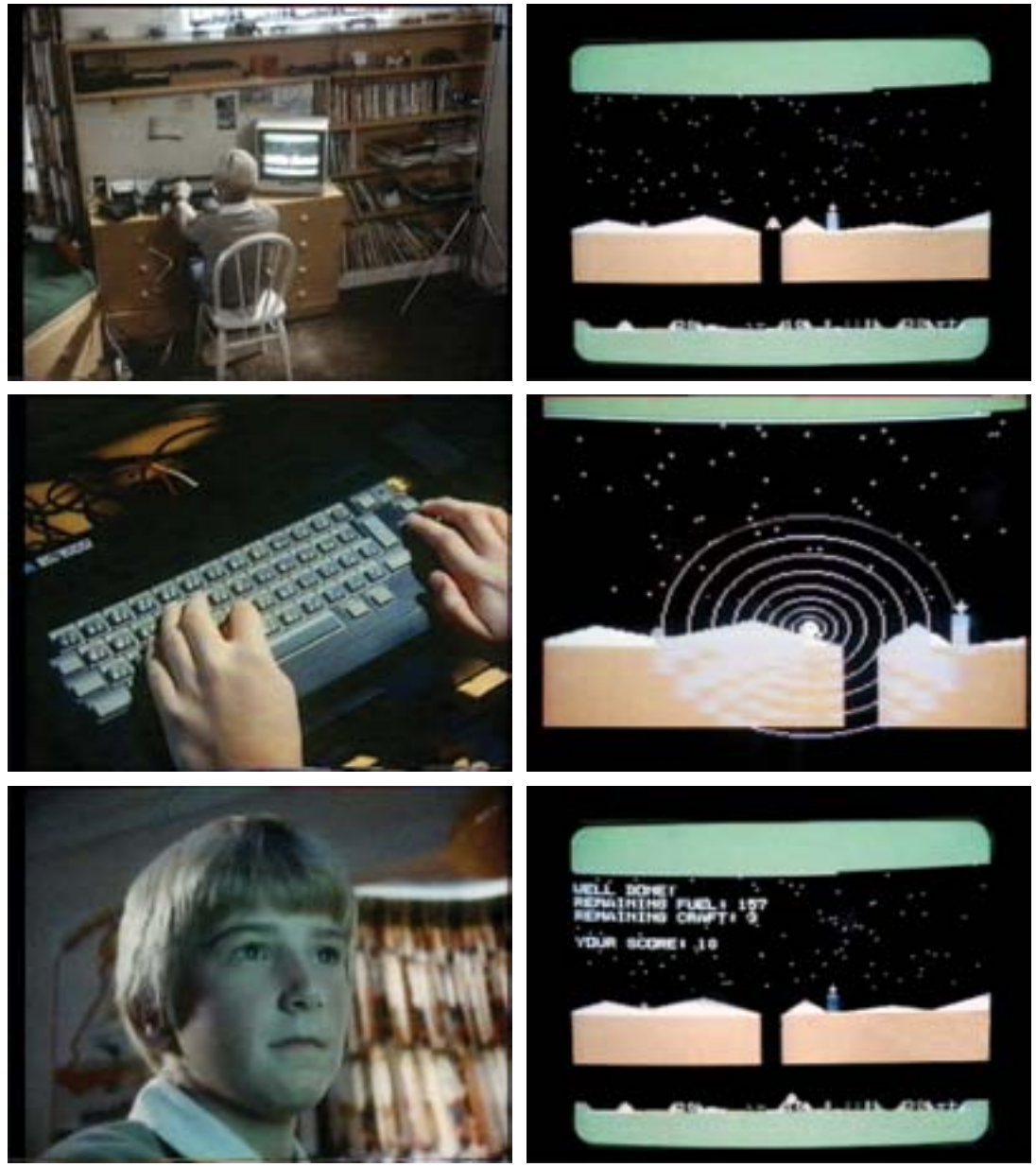
The computers of the 1980s demanded that users learn to program. Programming might therefore be thought of as a use that was *implied*. As Mark Sibly and Simon Armstrong explain, in response to my asking how they learnt to code:

> Mark: *It was much easier in those days, because you turned on a computer and you basically had to program. That was all you could really do. So, the first thing you had to do was command to load a program and then run it.*
> Simon: *And you've learned two commands, 'load' and 'run' and you go from there.*
> Mark: *And the manual that came with those early computers would have, in the back of it, a hardware diagram. It would have all the highly technical machine code stuff at the back… everything you would possibly need to know about it.*

This is in sharp contrast to contemporary computers: effectively 'black boxed', most people know very little about what goes on inside them.[8] As Sibly and Armstrong continue:

> *Mark: Nowadays, if you want to find out something about computers, you can spend a week on the internet and still not find it.*
> *Simon: It would be a bit like, if you have an old car, you can buy one of those books, and you can pretty much take cars to bits and put them back together again with one of those books. But there's no way you could do that now with new cars. And computers, it is the same, you can [sic] get a book that actually tells you everything, so you can teach yourself.*

4. Michel de Certeau, "General Introduction," in *The Practice of Everyday Life*, trans. Steven Rendall (Berkeley: University of California Press, 1984), xii-xvi.
5. Colin Beardon, *Computer Culture: The Information Revolution in New Zealand* (Auckland: Reed Methuen, 1985).
6. Swalwell, "The Remembering and the Forgetting of Early Digital Games: From Novelty to Detritus and Back Again," *Journal of Visual Culture* 6, no.2 (2007): 255 – 273.
7. Quoted in Pat Churchill, "Coming Up: More Home Learning," *Bits and Bytes*, December – January (1983 – 84): 63.
8. Dylan Evans, "Smash the Windows," *The Guardian*, November 6, 2003, np.



1: John Perry, *City Lander*, 1984, computer game.

fig. 1

Like Sibly and Armstrong, my other informants' learning was also largely informal. Typically, it was a process of trying and testing, trial and error.[9] John Perry described how he came to understand Basic as if he were "learning a new language." While his family had some books around, which he would occasionally refer to, more often, he recalls:

> *…you learn a 'word'—a function, a command—and you use that, and suddenly it changes all your programs, because suddenly you've got something that you can do. And then you learn some other trick. And generally I'd see someone use it in a program or something, and I'd look it up in a book sometimes, but mostly there was [no need for a book]. Mostly to start with you'd just copy the program and change a few little things to work out what was going on.*

Fiona Beals received her ZX Spectrum from an elderly couple whom her mother knew. They had decided it wasn't much use to them. The instruction book that came with it wasn't helpful, telling only how to set up the computer to type a letter. Beals found the Usborne range of books more useful, and, using these, she taught herself to program. To start with, she would type in other peoples' game programs. Then:

> *Once I clicked onto what was happening with the book, I was able to go 'well actually I don't want it to do this, I want it to do that.' I could start manipulating the code to do other stuff…*

Like Perry, hers was a process of learning by doing:

> *When I was writing the code myself, I would always write 5 or 6 lines of code and then end it, and test it, run it and see how it would go.*

Beals grew up in Westport, an isolated region of New Zealand, with little contact with others who were programming. As such, she recalls not knowing that what she was writing was a computer language:

> *I didn't know that until I learnt Basic in high school, and I was like 'Hey, I know all of this, I know this idea.' I didn't know all those things were actually a computer language. I had just thought 'oh, you have to give the computer instructions', and I actually clicked onto as a kid that the instructions were like a flow chart—that you could send it back and forth and stuff like that.*[10]

Magazines were another useful resource for the budding programmer. Typically, magazines of the period contained source code, as Katharine Neil explains:

> *You know how you buy games magazines now and they have CDs or DVDs on the cover, in plastic attached to the cover? In those days they had the source code, they printed the source code [in the magazine], in Basic. And you'd type it in. So they'd have a couple of pages of the code for a game, a sample, simple little game.*

In New Zealand, the locally published *Sega Computer*, *Bits and Bytes*, and *Computer Input* (known simply as *Input* to those 'in the know') were important for staying in touch with other coders. The contributions of readers/users to these magazines were many: subscribers sent in their programs, high scores, and the contact details of user groups in their area. As the editor of *Sega Computer* wrote encouragingly:

9. I draw here from Raymond Williams' meditation on the term 'experience' in *Keywords*. Noting the term's complexity, he writes that experience was "once the present participle not of 'feeling' but of 'trying' or 'testing' something," that is, of experimentation. Raymond Williams, *Keywords: A Vocabulary of Culture and Society*, revised ed. (London: Flamingo/Fontana, 1983), 128.
10. I have had a similar realisation during the course of this research that I, too, knew some Basic, though in my case, this was taught in computer classes at school in Australia.

> *It is important to note that this is YOUR magazine. So please send in any programs that you have, be them [sic] small or large, complex or simple…it matters not. If someone sends in a program and someone else learns from it then it has been worth it! To be quite frank I could name ten people in the UK, and three in New Zealand who now make a lot of money through writing programs, and they all started by writing a few simple programs and having them published in computer magazines! SO GET WRITING!* [11]

Magazines were also the place where the achievements of fellow programmers were celebrated, such as the 1984 publication by Grandstand of the 13 year old John Perry's game, *City Lander.*[12] And *Sega Computer* provided the enterprising Allan Rodgers of Gore—then 17—with a contact for Grandstand's in-house programmer, Michael Howard, to whom he wrote for advice on a career in programming.[13]

Some schools allowed pupils to take school computers home on weekends and over holiday periods, or to frequent the computer room at lunchtime, and these provided important opportunities to master coding techniques. Yet few of my informants credit formal school computer classes with igniting their interest in programming. Some gleaned knowledge from local users' groups, of which New Zealand had a considerable number. For their part, Sibly and Armstrong found these meetings too much "like church" and only attended a couple of meetings, preferring to spend time "at home or at friends' houses, learning how to program and making little games, experimenting with them."

Finally, computer camps, such as the ones planned by Peter R. Carr and Barry C. Small for the summer school holidays one year, offered another source of information and learning about computers. This series of camps aimed:

> *…to make the children aware of computers, the power and versatility of computers and the effect of modern technology on society now and in the future… teaching them to use standard application programmes and packages, teaching robotics, and voice synthesis methods and electronic and servicing techniques through building a simple computer. Other topics including vocational guidance, future developments, effect on society, covered by use of visiting lecturers and video training films.*[14]

Whether the attendees learnt to code or not isn't clear from the brochure. But like some user groups, one wonders how much fun the kids would have found some of the activities (i.e. were schoolkids of the 1980s interested in the societal effects of computers?).

Users as makers

The use of home computers, as the above accounts indicate, was fundamentally experimental. And this home use is one of the first moments when experimentation with digital technology is widespread. Home coders represent a classic case of use and consumption as a form of active production. In *The Practice of Everyday Life*, De Certeau makes a distinction between production and the uses that are made of products by users who are not producers.[15] His work is useful for understanding the active form of consumption that is experimentation: the verb '*faire*' in De Certeau's French title *Arts de Faire*, communicates more

11. *Sega Computer,* "Editorial," *Sega Computer,* Jan/Feb/Mar/Apr (1986): 1.
12. *Sega Computer,* "Sega's Young Programmers: Today New Zealand, Tomorrow the World," *Sega Computer,* 1 (1984): 21.
13. Allan Rodgers, email message to author, 16 November 2007, citing letter from Michael Howard to Allan Rodgers, 3 December 1985.
14. *Hey! These Holidays you can go to the 6 Day 'Computer Fun Camp'* [brochure]. Auckland Collection: City Library, New Zealand Department, Ephemera/ Computers, no date.
15. De Certeau, "General Introduction," xii-xvi.

effectively the sense of an active making (in French, *faire* means to make or to do) than does the English term 'use', which tends to imply functionality and instrumentality.

Writing games at home was undoubtedly one of the most common *uses* that were made of early home computers. Fiona Beals describes one of the programs she wrote:

> *The one program I do remember, which was totally fresh, was that I generated a pyramid on the screen that was firing bullets, simulated style, probably numbers and letters (I used to use x's to generate the picture). So I had this pyramid of x's, bullets that were x's, and aeroplanes that were flying across, and it was all random. When the bullet hit the aeroplane, the aeroplane would explode, but it was done in a randomised way, so that you never knew where it was going to hit. I remember that I left it on because it was such a great thing, and my computer overheated.*

Katharine Neil's home coding labours produced a tic-tac-toe game. Neil recalls:

> *I didn't know anything about graphics, so I wrote a tic-tac-toe game in ASCII, because I didn't know …no, I knew how to draw lines. I drew lines and I used ASCII characters. I thought it was very, very clever, because I thought my computer AI was very, very clever, because the computer would always draw or win. It was impossible for my computer to lose. And I thought that was very clever for a few years, until I realised that everyone knew that!*

The majority of home coders were amateurs, whose coding remained a hobby. Indeed, all of my informants were still at secondary school during the 1980s (subsequently, a number would have careers in various aspects of software development). There were, however, opportunities for home coders to gain *some* type of advantage from their programs. As already mentioned, magazines talked up the possibility of hobbyists turning professional, and the advertisement placed by Dick Smith Electronics in the December/January 1983 – 84 issue of *Bits and Bytes*—advocating "enterprising computer buffs" to profit from their hobby by writing programs for the new Dick Smith Colour Computer—illustrates this interest in the programs of home coders. Some coders, such as the young Mark Sibly, were thus able to benefit from their hobbyist involvement, albeit in a small way. Sibly "sold" his Vic-20 game "Dinky Kong" to an Auckland computer store owner, who later commercialised it in the U.S. His return from the transaction was an external floppy drive, enabling him to save his programs to removable storage media.

There is, however, a degree of instability or slippage around this apparent hobbyist mercantilism. I have heard a number of stories from people who aspired to sell their games, and were not able to. For instance, after having the code for his game "Harbour" published in *Computer Input* and selling "City Lander" to Grandstand for $300.00 in the same year, John Perry wrote another game which he also showed the company.[16] Grandstand, however, were not interested, as they already had a title in "Dungeons Beneath Cairo" that was similar to the one he was offering. "Dungeons Beneath Cairo" was considered superior because it was written in machine code rather than Basic. Others, like Andrew Kerr, who intended trying to have his untitled machine coded game endorsed by Poseidon software, never got around to it ("but it did get me an 'A' in computers

16. John Perry, "Harbour," *Computer Input*, July (1984): 21 – 23.

EDUCATION

Computer Age Learning

By Pat Churchill

There is no disputing the magnetic attraction of the computer. Wherever a computer is on public display going through its paces you can almost guarantee the surrounding few metres will soon be packed with people wanting a look, or better still a go at the keyboard.

The Learning in the Computer Age Exhibition at Wellington's Michael Fowler Centre in October was no exception.

Organised by the Wellington branch of the New Zealand Educational Administration Society (NZEAS) the exhibition featured the latest in computer software and audio-visual educational materials, plus a comprehensive range of computer literature. It was very much a "show and tell" affair with an opportunity in the afternoon for teachers, students, and parents to look over the displays.

Exhibitors showed educational software for a variety of computers including the Poly 1, Atari, Commodore, BBC, Apple, ZX81, BMC, and Pencil II. There were also packages for teachers and school administrators covering such topics as time-tabling and course options.

The Wellington Polytechnic's Poly 1 display attracted considerable interest, and an adjacent display showed how physically handicapped children at Kimi Ora School in Wellington are able to use a Poly 1. The bright yellow machine featured three large keys to enable users to tap out letters in Morse code — a dots key, a dashes key and an "enter" key. This was just one way a disabled child could operate a computer.

One interesting feature was the number of youngsters introducing their parents to computers. The exhibition will no doubt generate another round of cake stalls.

As Dr Colin Boswell, president of the New Zealand Computer Society, told a local newspaper covering the exhibition, "cake stall money" has provided more than $1,500,000 towards computers for New Zealand schools — more than the Education Department has allocated for the same purpose.

Dr Boswell, director of the computer services centre at Victoria University, said the Education Department had established a two-person curriculum unit to make programs available to schools, and had identified five preferred computers for secondary schools, but it needed to do much more or New Zealand would slip behind the rest of the world.

Parents viewing the exhibits were certainly enthusiastic and keen for their children to have access to computers at school. Some were clearly interested in the display with a view to purchasing home computers.

Dr Lynette Hardie Wills, of NZEAS, said the society was delighted with the response to the exhibition. A series of musical activities was organised in conjunction with the exhibition and participating youngsters were encouraged to bring their parents to view the displays as well as to hear the music.

"In addition to having parents and teachers present, we also wanted members of the education community who are making decisions about buying materials to come along. And we invited school councils and representatives from the Education and Labour Departments.

The Department of Education had been very supportive, she said, and had mounted a striking display.

Dr Wills said the exhibitors, too, were delighted with the amount of interest shown.

"There was a lot of informing going on."

at college!").[17] In part, too, the energy and experimentalism of home coding ensured it would remain on the margins of 'legitimate' production. Despite the importance of games in driving the uptake of early computers, 'serious' software people (such as those who wrote software for mainframes or for accounting purposes) adopted a kind of 'snobbery' towards the user end of the market: fearing the association with games would diminish their credibility, Neill Birss recalls they wouldn't advertise in *Bits and Bytes*. Whilst productive, then, in most cases home coding remained a practice that "produce[d] without capitalising."[18]

A computer in the 1980s was, as Katharine Neil puts it, "an expensive toy":

> *[People] use computers for work now. You didn't used to use computers for work. People with home PCs use them for writing their CVs, and doing work-y things like using them as a communication portal and all that. Whereas computers in the 80s, when I was a kid, a computer was a luxury item for having fun… it wasn't something that you had in your house because you needed a computer, like you need a mobile phone or something, and you do a bit of work on it as well. It was definitely a fun toy… far more of a toy than it is now. Today, people are like 'Oh, I spend so much time in front of a computer, I go home and I don't want to spend any more time.' Whereas back then, it was like some cool novelty—it was like a Ferrari or something. It was like a Sunday drive: it was a luxury car that you drove around in on a Sunday, not a commuter vehicle that you sit in traffic jams every day in.*

Home coders' involvement with the computer as a "fun toy" laid the ground for an everyday, creative engagement with computers: a *messing around with computers*, a seeing what was possible. Theirs was a curiosity driven experimentation, in the tradition of making crystal set radios and hotting up cars, according to Neill Birss. Reflecting on teenagers' ready embrace of the challenge of coding, Birss ponders whether or not there was a peculiarly New Zealand dimension to this:

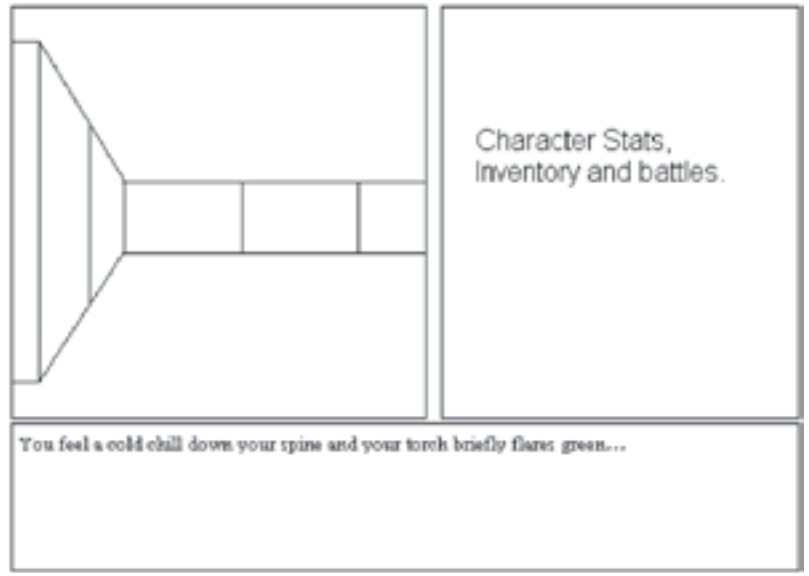
> *I think perhaps when you're very remote you don't have the sense that hey, this is a big thing. Maybe if you grew up in Silicon Valley and you thought about writing something you'd see all these huge guys and you might be a bit more daunted in some ways.*

While New Zealand's remoteness is sometimes a popular 'cultural narrative' for the creativity of its inhabitants, I am more interested in thinking through what home coders did and what the significance of their activity is (rather than *why* they did what they did).[19] Home coders' ways of using the computer are particularly compelling. In the 1980s, programming seemed to involve an element of play, whether users were writing games or something else. This playful approach is distinct from the common instrumental relation to technology—so well diagnosed by Martin Heidegger—where the computer is viewed (predominantly) as productivity tool.[20] The persuasiveness of Heidegger's critique of the instrumental conception of technology, and the status of his essay, "The Question Concerning Technology," as one of the seminal texts in the field, however, also acts as a limit, in that it contributes to the naturalisation of instrumentality; thinking and theorising other relations to technology, beyond the instrumental, becomes difficult. Home coders' provide a contrary example. Their use of early computers was not instrumental: it could not be, as they often did not know

17. Andrew Kerr, email message to author, 15 November, 2007. There is more detail on the plot and execution of Kerr's role playing game in Swalwell, *Early New Zealand Software Database*, 2007. http://www.nztronix.org.nz/main.php. That Kerr provided a mock-up—rather than an image from the actual game, or the game itself, which he writes was "inadvertently lost"—points to the fate of many of the products of home coding. Similarly, when John Perry sold his computer, the "stacks of games" he'd written went with it. As well as collecting information about the range of software writing activity that went on in 1980s and 1990s New Zealand, the aforementioned database endeavours to collect information about such (lost) hobbyist titles, by inviting members of the public to contribute what they know via an online form.
18. De Certeau, "General Introduction," xx.
19. Brian Sweeney and Kevin Roberts, NZ Edge: 1998 – 2007, 2007. http://nzedge.com. See also Jan English-Lueck, "Number Eight Fencing Wire: New Zealand Cultural Innovation and the Global Silicon Network," adaption of a poster presented at the Annual Meeting of the American Anthropological Association, Chicago, November 30, 2003. http://www.sjsu.edu/depts/anthropology/svcp/pdfs/svcp8wir.pdf
20. Heidegger, "The Question Concerning Technology," in *The Question Concerning Technology and Other Essays*, trans. William Lovitt (New York and Cambridge: Harper, 1977), 3-35.

what they were doing. Effectively *making it up as they went along*, home coders invented a new relation with the computer. By their own accounts, it didn't seem all that remarkable at the time, partly because such use was widespread and partly because relations with home computers had not settled into any *one* particular pattern.

Some contemporary digital makers—artists and others—embody an approach to computers that is reminiscent of home coding, an *ethic of experimentation*, in the sense that home coders' trying and testing, their probing of possibilities, provided a base for invention.[21] Having little regard for orthodoxy, such an ethic is both pragmatic, and displays a certain daring in following a line of thought to see where it leads and what might happen. The result might be unexpected and is often unlikely. As with home coders, what such acts of *poiesis* do is to bring forth other possibilities, new relations to technology, other modes of revealing.[22]



Andrew Kerr, 1988, mock-up screen view of untitled role-playing game for the Sega SC3000.

21. I am not implying that artists do not know what they are doing, though not knowing what one is doing, on occasion, is to exist in a state of grace.
22. I am using *poiesis* here consistent with De Certeau's quite broad usage, to mean creation, generation, invention rather than Heidegger's (somewhat narrower) reference to the bringing-forth of the fine arts and poetry. See De Certeau, "General Introduction," 205 n.2; and Heidegger, "The Question Concerning Technology," 10.