
A History of Llamasoft

by

Jeff Minter

and

Patrick Minter

Copyright Llamasoft Ltd. © 2009

All Rights Reserved

And All That Bollocks.

Presented on the occasion of the release of “Gridrunner Revolution”. This version of the History is a work in progress. I hope you enjoy reading it.



Cheers!

- Yak, August 2009

Table of Contents

Part 1: In the Beginning.....	7
Part 2: Colour, Sound, Poking Around.....	20
Part 3: Welcome To The Machine.....	31
Part 4: The Joy Of Hex.....	44
Part 5: Fur-ther Education.....	59
Part 6: Higher Learning.....	71
Part 7: A Beginning.....	86
Part 8: The Dawn of Llamasoft.....	105
Part 9: Mutant Camels.....	128
Part 10: Go West, Young Man.....	140
Part 11: An English Country Garden.....	150
Part 12: PESCLR.....	161
A History Of.... Gridrunner.....	173
Visual Synthesis.....	195

Part 1: In the Beginning...

The first lesson I was ever taught about computers came from my dad. He and my mum were both students of the Open University, which for me as a kid was brilliant. They were both studying science units and would get sent experiments to do – my dad's usually involving something to do with physics, while my mum's were biology-related. So, my dad would get sent all kinds of interesting gizmos with which he would set up experiments on the dining room table, and I got to go and watch him rolling balls down inclines and doing interesting things with masses and springs and measuring and timing stuff. My mum would have loads of intricate text-books I could read and occasional treats like a sheep's brain to dissect (which I could then take to school and gross my mates out with when she was done with it).

I remember I must have been about nine years old when my dad had a computer science unit on his course. I knew he had to go to a

technical college near Reading to do the course, and that he was studying something called “Basic”. I had been brought up on a diet of Asimov and Clarke, and to me, computers were things like Asimov’s Universal AC – electronic ‘brains’ filled with all the knowledge of mankind and eerily sentient. The idea that one only had to go as far as just outside Reading to commune with one of these entities intrigued me. One day, I was hanging out in the kitchen in a nine-year-old kind of way, when my dad happened to come back from his course. I was moved to ask him: "Da-aad, if there are so many problems in the world, why don't we just ask the computers what to do?"

He explained to me that computers weren't like the machines I'd read about in my sci-fi books at all. In fact, they couldn't ‘think’ at all. The only things they ‘knew’ were things that you had explicitly told them yourself. He showed me bits of paper with some printed output on, and I saw the results of his dialogue with a machine – as it asked for some numbers and did some simple arithmetic on them, and played a game involving taking matches from a pile. Mildly interesting, but hardly world-saving stuff. I promptly forgot all about computers for almost another nine years.

1978. I was sixteen, and had just left secondary school to go to sixth-form college. I'd come through with a reasonable crop of ‘O’ levels – nothing particularly great. My best subjects were English, Physics and Maths, and, accordingly I joined the sixth-form to do ‘A’ levels in Maths with Mechanics, Physics and English Literature – something of an odd combination. I still really didn't have a clue what I

intended to do once out of education – just some vague idea that it might be something to do with physics, since that was what my abilities seemed to dictate. English Lit seemed particularly out of place in a system which tended to incline the arts and sciences to separate.

At the start of sixth-form we had to sign up for what were rather misleadingly called "main studies" - misleadingly, because they weren't your main studies at all, but little bits of skiveable mini-courses that were used to fill up the curriculum between the main 'A' level studies. I remember talking to my head of form about what to choose for these, and he suggested, amongst other things, "Computing Main Study".

"What, you mean using terminals and such?" I asked him, vaguely remembering my dad's sheaf of teletype output from many years before. He replied yes, and although I still had no particular interest in the subject (my futuristic idea of what computers were having been replaced by a more mundane belief that they were good only for sending out bills and Reader's Digest junk) I thought I'd give it a go anyway. It had to be better than needlework or cookery or somesuch, and, if I didn't like it, it was a potentially skiveable 'main study'. So I signed up for Computing Main Study.

Of course, I would like to say that this was the spark which kicked it all off, and that this wise and insightful suggestion by my form head led to the ignition of a passion that was to burn clear and undimmed across a couple of decades. Alas, that wasn't the case. I think I only ever stuck out about two lessons of Computing Main Study. The content of the course was the trouble. It was as if someone had sat down (or quite possibly a large group of people, the kind of people who enjoy sitting on committees and probably end up being

town councillors) to devise a method of teaching possibly one of the most exciting disciplines in the world in the most boring and off-putting way imaginable.

Terminals? Even the clunkiest old Teletype chuntering along at a few characters per second would have been racy and thrilling compared to what was taught in Computing Main Study, because at least that would have been interactive. The genius of this course was that it managed to completely remove any notion of interactivity from the computing experience. Perhaps it was a throwback to times when even Teletype terminals were rare...

On the first lesson, we were told that we would be learning a language called CESIL. This wasn't any kind of a real language that anyone used to really do things with, from what I recall, but some synthetic language purely for the purposes of education (or perhaps places on real comp.sci courses were already getting oversubscribed, and CESIL was deployed to send lesser students running gibbering and screaming into the hills, vowing never to go near a drop of code ever again). During that first lesson, we were taught the keyword for 'Print' (and I can't remember what it was, but I see no reason to think that it might be anything other than 'Print') and the appropriate syntax for using it. Then it came time for us to write our first CESIL programs.

We were all handed sheets of paper with boxes to write the letters of the program in. Upon these sheets, we had to painstakingly write the two or three lines of our simple programs, in capital letters, in the little boxes. Once this was done, the sheets were collected up and put into an envelope by the teacher. Then, we were told, they would be sent by mail to somewhere else, where the programs would be typed in by operators (who would slavishly copy EXACTLY what you wrote,

including any syntax errors) and run as a batch job on some system, which would then produce printed output, which would be sent by mail back to the college for us to finally look at the following week.

I remember I did go to the second lesson, just out of curiosity, to see if what I had submitted had run correctly (it had – a couple of people must have made errors with the syntax because they got error messages) but basically I thought enough was enough. If computing meant waiting a week to get a couple of lines of output or a syntax error message then it wasn't for me. I prepared to do some judicious skiving at appropriate times and just concentrate on my key 'A' level courses.

That might have been the end of it except for one thing: videogames. I'd seen a few by that time - one of my brothers brought round a Pong machine (I remember it was the latest thing and had cost him forty quid – a fair whack back then) and it had seemed like magic when he'd plugged it into the telly and we sat down to play. I'd watched that same brother break through the wall in Breakout whilst under-age drinking in a smoky wine bar in Basingstoke the previous Christmas. I'd even had my first encounter with Space Invaders by then, in the arcade tent at a travelling fair that came every year to the town where I lived. Still, there was nothing in my brain that connected videogames and computers. Games were just black boxes that someone had built to do the things they did by way of some kind of incomprehensible magic, and computers were things that sent you your gas bill and, apparently, took a week to send you an error message if you got things wrong.

One day, not long after college had started, I happened to wander into a room where I saw something that interested me. A dark-haired lad was sitting in front of a machine which looked like a cross between a telly and a calculator, and on the screen there was something that was crudely but recognisably a videogame. He was pressing keys on the keyboard and a blob was moving around on the screen. I watched over his shoulder for a while, and then I asked him "How did that game get in there, then?" His reply changed my life in an instant...

He turned and said: "I typed it in".

He stopped his game and typed LIST on the odd little calculator-like keyboard of the machine and lines of code flowed up the screen. He explained that this was BASIC, that the machine was something called a 'Commodore PET', and that if you learned a bit about BASIC, you could make games happen on the screen. I was fascinated. The idea that you could type in games – even make up your own... I suppose that's the point where my real education began.

I went straight to the college library and found a book on BASIC. On the bus-ride home from college, I pored over the book and began to get the first inklings of how it worked - it was a bit like algebra, but not quite. The statement ' $a=a+1$ ' is nonsense in algebra but was an instruction in BASIC. The word 'string' came to mean something other than what tramps tied their trousers up with. I'd used programmable calculators before, so the concept was not entirely alien to me, but compared to awkwardly programming thirty steps of purely

mathematical operations with only a row of LEDs for output, this was a whole new world.

By the time I got off the bus, my head was spinning. I thought I understood a few BASIC basics – like how to use variables and what a FOR-NEXT loop was and what IF-THEN did... I was still no closer to being able to make up games with it, but I wanted to do something. My brother was working for the AA at the time and had the loan of a TI-59 programmable calculator that came with a booklet of example programs. When I got home I went to his room, blagged the booklet and looked through it to see if I could make use of one of the examples as a starter for my first bit of BASIC code.

My parents must have been surprised to see me nip with such alacrity off into my room with books in hand - I wasn't known for being keen on homework. I flipped through the calculator programs to find something that looked simple, and sat down to write - in biro on a bit of paper - my first bit of proper code – porting a biorhythm program from a calculator language that I'd never used, to a computer language that I'd never used.

The next day, my parents must have been surprised again, because I actually got up early to go to college. This never happened – ever – so it was the cause of quite some shock when it soon became a habit. On this first day I arrived in college before anyone else and made straight for the Commodore PET. I fumbled around the unfamiliar device looking for the power switch, turned it on and saw for the first time something that was to become very familiar:



Commodore PET 8k boot screen.

...with a flashing square underneath prompting input. No writing stuff on forms and sending it off for a week to write code on this beast. I fished out the bit of paper with the few lines of BASIC code I'd written and started to enter them into the machine, putting numbers at the start of each line as I'd read one should.

For the first few weeks there was novelty even in the act of typing itself – and seeing letters appear not on paper but in glowing phosphor on the screen. For most of my life, the TV had been entirely a one-way device, and only relatively recently had the arrival of videogames started to change that. To actually be able to write on a screen in this way seemed strange and rather cool.

I entered my few lines of code and typed RUN, and it worked - almost. The machine asked me for my date of birth as expected, but upon printing out the state of my biorhythms, something was wrong – unless all my biorhythms were flat lines. Luckily, at about that time, I was joined at the machine by the dark-haired lad from the day before – who I would soon get to know as Mark 'Ruptured' Rawlinson. He was

on the proper computer science ‘A’ level course, and pretty much Head Geek on account of having had experience on a RM380Z system at his previous school. We went over the code and he helped me fix a couple of errors. Then, we took it in turns to find out the state of our biorhythms that day.

Over the next couple of weeks, I settled into geek society. I got myself a cassette tape for storing my programs on, which I carried everywhere with me. I met the other core geeks - Mark ‘Clovis’ Clover and Andrew ‘Mole’ Round. Lunchtimes and breaks were spent clustered around the PET, either coding or watching someone else coding. We’d be first into college in the morning and last to be booted out by the cleaners at night. There was no manual with the PET, so we learned what we could by experimenting or from the occasional snippet of information gleaned from obscure magazines. Out in the real world, the rest of the students did their things just like in any college – the drama freaks and the science lot, the cool people and the nerds... but we were blissfully and uncaringly ignorant of all that. We worshipped in the glow of that little screen.

I continued to tinker with my little biorhythm program and in the process began to learn that not all BASICs were equal. The library book (which I had decided wasn’t going back to the library) described a very generic BASIC that would work on just about anything, including teletype terminals. Commodore BASIC had extensions which let you do things you could only do on a screen, such as position the cursor anywhere before you PRINTed something. I played around with that and eventually had my biorhythm program actually graph out the three intersecting sine waves over a period of time. The PET also had

‘graphics characters’ – special shapes not in the normal alphabet which you could use to embellish graphic output.

There was another trick you could do with PET BASIC. As I mentioned, there were commands to move the cursor anywhere on the screen. You did this by printing special characters - <HOME> would send the cursor to the top left of the screen (position (0,0) in the character matrix). If you then printed ten <CURSOR DOWN> characters and two <CURSOR RIGHT> characters then you knew the cursor would be at position (2,10) in the matrix - that was how I’d got my biorhythm graphs to appear.

The neat trick involved moving the cursor relative to its own position. Say you started in the middle of the screen, and printed a ‘*’ there. After printing a character, the cursor always moved one position to the right. But you could control the cursor position with special characters, and do this: print a <CURSOR LEFT> character, which would place the cursor ‘on top’ of the ‘*’ you already printed there. Then print a space, which would delete the ‘*’ and move the cursor back over to the right. Finally you could print another ‘*’. Given the speed of execution, doing that gave the appearance of causing the ‘*’ to move one character to the right.

From there, it was a simple step to create a string of three characters - <CURSOR LEFT>, a space, and a character of your choice - the ‘graphics character’ which looked a bit like a ball, for example. Call it R\$. Print it repeatedly... and you had motion. The ball would appear to move of its own volition across the screen – just like the balls in Breakout and Pong...

Other strings would move the ball in different directions - L\$ for left, that was <CURSOR LEFT>, space, <CURSOR LEFT>, <CURSOR LEFT>, ball. U\$ for up was <CURSOR LEFT>, space, <CURSOR UP>, <CURSOR LEFT>, ball, and so on... IF-THEN let you work out if the ball was at a screen edge and change its direction accordingly. There was even a function, GET, that let you see if a key was being pressed on the keyboard, and allow you to change the direction of what you were moving accordingly.

You could move objects, and they could be aware of their environment, and they could be controlled. At last I could see how to get to 'game'.

We knew how to get to game outside college, too. Since I'd (unofficially) packed in Computing Main Study, I took up Space Invaders Main Study, instead. Being sixth-form, matters of skiving were much more civilised than in school, and although technically there was a dress code, it wasn't really a uniform, so one could pass off as a non-student in town easily.

It became second nature to us to stroll out of the college gates, under the bridge, down the alleyway emerging at the top of town, perhaps having a Grigg's pie and buying a copy of PCW or Practical Computing at the newsagent along the way, then down the hill and over the overpass into New Market Square. Here there was a record shop, Harlequin Records, with a bright orange sign outside and a Space Invaders machine lurking in a dark recess. We'd spend quality illicit time here, taking turns to play or watch each other's games. A hardcore of skivers from other educational establishments would aggregate here,

too. Occasionally, I'd buy a Gary Numan single or something just to placate the poor shop owner, because we spent a lot of time in there but seldom bought anything - we just wanted to play.

When I first started going there, I'd only had a couple of fleeting encounters with the game and I was not at all expert. I remember watching in awe some kid who was nonchalantly manoeuvring in the open space under the rack of Invaders, casually dodging bullets, instead of cowering under the nearest house and squeaking out every now and again to take terrified pot-shots (as I did). We learned through practice and osmosis, watching each other's games intently when not at the controls ourselves. I remember the sheer joy of finishing off my first ever sheet - the heart-pounding intensity of the last Invader running like the clappers down the screen, the carefully-timed shot, the satisfying "ZAP!" followed by a blissful moment of silence and exultation before the next rack appeared and the game's slow, steady heartbeat resumed... It was like a little orgasm.

I became quite competent at Space Invaders – but not yet awesome. Before I was to encounter terrifying techniques such as the Death Row Execution Method I'd need to study the methods of one of those true videogame masters one occasionally encounters - annoying people who are instantly good at any game they play and achieve awesome skills if they actually bother to practice (our own Sickboy being a prime example). I wasn't to meet one of these individuals until a couple of years later at University though, and for now we settled for competence, learned to handle ourselves without fear in the war zone... to pick off enemies in such a way that we controlled their advance down the screen... to count shots so that we always milked the

maximum 300 points out of the little saucer every time - the basic skills of a solid Space Invaders player.

Us geeks from the sixth form saw the game with new eyes, too. We knew that as the objects on the screen moved, inside the machine values were being incremented and decremented to make them move. We knew that position comparisons were being made to detect when objects collided. We knew that behind it all, lines of code ticked away – orchestrating, regulating...

And we wanted some of that action.

Part 2: Colour, Sound, Poking Around...

Back in college, we finally had enough knowledge to begin making simple games. And simple they were – to the point of being ludicrously primitive by any reasonable standards. But they were our games that we had made ourselves, and there was something immensely satisfying about playing a game with your mates and knowing that it had come entirely out of your head.

I remember writing a realtime version of those Star Trek games that were popular at the time. In mine, instead of being a turn-based game, you guided the ‘Enterprise’ (a PET graphic character that was meant to be a playing-card club symbol) across sectors of the galaxy in real-time using the keyboard, one sector per screen. Sometimes, you would encounter a ‘Klingon’ or a ‘Romulan’ (cue more PET playing-card symbols) and a firefight would ensue, with everyone firing straight and diagonal line characters at the other (‘lasers’).

Out of that grew a two-player dogfight game, in which each player controlled their 'ship' from opposite ends of the keyboard (a bit cramped, since the PET keyboard was tiny) and bumbled around, sniping at each another (one of the ships was just an X-shaped graphics character, so of course whoever had that was said to have the X-Wing). Ahh, you needed real imagination to flesh out your games with a little atmosphere in those days...

We spent every spare moment in front of the PET, coding up games, watching others coding up games, or playing each other's creations. In due course, the PET was joined by a TRS-80 Model 1, which became known to us as the Trash-80. This was a Z-80 based machine, and although we were curious about it at first, it didn't appeal so much to us gamers - it lacked the PET's 'graphics' characters and the keyboard, although at first sight appearing to be nicer (being actually keyboard-shaped, rather than small and weird and fiddly like the PET's) was a bit tacky and prone to double-entering. We stuck with the PET and carried on creating and played our little games.

I remember that before long I began to run into difficulty as the ambitions for my games got loftier. Everything was fine as long as there weren't many objects on the screen and the environments weren't too complex. But I wanted to have many enemies on the screen at once, and gameplay arenas more complex than just a straightforward box. But BASIC was too slow and unwieldy to do some of the things that needed doing, and although I had achieved competence at BASIC programming, I really had no idea how things actually worked inside the heart of the machine.

'Ruptured' Rawlinson was the guru of the geeks, and he had told me about some strange BASIC commands called PEEK and POKE. At first these commands seemed completely mysterious, because I had no idea what it was they actually did. I knew it was something to do with accessing parts of the machine's inner memory, and we used them mainly to try and make strange things happen by altering values in a location called, mysteriously, 'zero page'.

We gleaned from magazines that BASIC stored some working variables in this zero page area, and by fiddling around and changing those values you could do odd things - like speeding up the flash rate of the cursor until it became a blur, or reducing the keyboard auto repeat rate so that the merest touch of a key would spew out loads of characters all at once. Sometimes, messing with these values did nothing at all, and sometimes it crashed the machine solidly. I can't say that I used PEEK and POKE on zero page to any great end, or really even knew what I was doing at all - it was all very much just voodoo... POKE about and see what happened, without really understanding why.

The problems with my games were starting to irritate me, though. I was working on a game where I wanted there to be some kind of structured, maze-like environment for the player to move around in, and it was all getting silly - I was considering dimensioning a massive array as big as the screen, and for each character I printed on the screen, storing a corresponding value in my big array for the purposes of collision checking. PET BASIC let you put a character at any position on the screen, but there was no command to let you see what was at any position. Hence my big, clunky, unwieldy array, which was as inelegant to use as it sounds. I was getting fed up with it... And then,

I thought: "Well... The system has memory of its own – I know that from using PEEK and POKE. So, somewhere it must have memory for where things go on the screen...".

I sat down and proceeded to search for that memory – by using the POKE command. It took two parameters (POKE X,Y) and I knew from using it to tinker with BASIC variables that 'X' was a system memory 'address' and 'Y' was the value you wanted to store at that address. I knew that there was 8K of memory somewhere in the PET, and I figured that some of that memory must be used to hold a copy of what was displayed on the screen.

I POKED here, I POKED there – at first randomly, and eventually more systematically, attempting to write values in locations that fell on 1024-byte boundaries, since it seemed logical to me that system memory might be laid out according to the size of 1K memory blocks. Eventually I typed:

```
POKE 32768,0
```

...and an '@' symbol appeared at the top left-hand edge of the screen. To confirm that I had found what I thought I'd found, I typed:

```
FOR X=0 TO 255: POKE 32768+X,X: NEXT
```

...and the entire PET character set appeared in neat order at the top of the screen, without a single 'PRINT' statement being necessary at all.

I had discovered the existence of screen RAM, and in doing so taken my first step towards real understanding of how the machine really worked, and away from the limitations of BASIC.

This was immediately absolutely superb for games. No more messing around uttering strings of cursor-positioning characters - you knew the screen was forty characters wide, you knew that location 0 of the screen was at 32768, and so placing a character with code 'N' at location 'X,Y' was as easy as `POKE 32768+(Y*40)+X,N`. And best of all, you could find out what was at a particular screen location by using the opposite PEEK command, so there was absolutely no need for the clumsy arrays I had been playing around with. It all simplified down to moving numbers around and being able to look at what numbers were already there.

This led to a whole new set of games we could write, with complex environments and lots of enemies on screen at once. One of the games we made was called Rhino, where the player had to manoeuvre for as long as possible to avoid enemies that constantly took steps towards him, in a vaguely Robotron style (the advancing 'Rhinos' were represented graphically by using the PET's Pi character). Another involved trying to bounce a ball towards a target using diagonal lines as 'bats' - you could have hundreds of bats on the screen at once and the ball would deflect through them all. We even arranged for a little bit of extra chaos in the gameplay by having the bats flip to their opposite state whenever the ball (PET character #81, bless it) hit one. The trajectories thus generated were complex and interesting. We called that one Deflex.

Our code, of course, became incomprehensible. The price of this new power was readability. When things had been done using PRINT statements, it was at least fairly obvious what we were doing. Now, our code was just control structures and lots of arcane PEEK and POKE statements. Knowing how baffling PEEK and POKE had been to me before the Great Glorious Moment Of Understanding, I know our PET code of that era would have looked baffling indeed to any normal BASIC user. We were becoming machine-specific.

My social life, such as it was, now revolved totally around geekdom. Every morning I would get the early bus to college (or, if the weather was nice, cycle the eight miles or so on my bike, listening to music on a crappy mono tape deck I'd bought from Dixons, along with a pair of new-fangled 'Walkman' headphones). Park bike, lock bike, make a beeline for the computer room, settle down to bang out some code, listening to Gary Numan on the crappy mono tapedeck. Eventually the others would show up – Mole, Rup and Clovis – and our normal routine of coding and playing would resume.

We had the machine to ourselves in the early morning and after college at night, but during the day, access to the machine was determined by a timetable. Every Monday morning there was a stampede of geeks as the week's timetable was posted up in the computer room – everyone could claim two half-hour slots per day, and it was important to get there early to claim the prime slots for oneself. I once bumbled straight into a teacher whilst I was running to get to the booking sheet, and received the worst punishment I ever had at college – banned from using the computer for a week! I was bereft.

Most lunchtimes when we weren't coding we'd bunk off and go down town, either to play Space Invaders at Harlequin, or to find games available elsewhere – upstairs in Woolworth's was good. In the electrical department could be found loads of tellys upon which Terry Griffiths was winning the Embassy World Professional Snooker Championship, and laid out amongst them were the game consoles of that era – bright orange Binatone 'Pong' games, SIMON, Merlin, an oddity called Star Chess... We'd loiter there buying nothing and fiddling with the tech.

Downstairs was the record department where every now and again I'd buy the latest Gary Numan single. There was a slightly serious little computer shop up at the top of town that we once went and poked our noses into. They had a PET in there and I remember loading up one of my games – a version of the same Star Chess game that I'd seen in Woolworth's and thought it might be fun to code.

The bloke in the shop asked me if I'd ever thought of selling it – which I never had, of course. Our hobby was pretty much unheard of out in the world at large, and the only others I knew who were into it were part of our little gang of geeks. We made games for each other for the fun of it and to learn, and there was no more thought of making money from it than one would think of making money from building Airfix kits or going fishing by the canal on a Sunday afternoon. We used to joke and wonder if one day in the distant future you'd be able to buy computers in Tesco like any other household commodity, but we didn't seriously believe it would ever happen. What we did was just too obscure. So we thought...

Weekends were a forcible interrupt to the coding flow. I would dearly have loved a computer at home, but these were pre-Uncle Clive days, and a PET would set you back a good six-hundred quid – far too expensive for a schoolkid with no income like me, and my parents certainly weren't rich enough to buy me one or interested enough to buy one for themselves. In fact they thought of my passionate interest as just another passing fad – like most kids, I'd regularly had intense periods of interest in a subject only to move on to something else a few months later.

My fellow geeks all lived in or near Basingstoke and I was out in Tadley, so, during these computer-detached weekends, I amused myself in other ways. I'd go out for long walks with my non-geeky mate James Lisney (the Baughurst Piano Wizard), usually ending up at a country pub for a quiet bit of underage drinking. Or I'd go down to Southampton to see my gran, and have a bit of a mooch around town there, fiddling with the Pong machines in the shops, keeping an eye out for Space Invaders machines to hone my skills on, going to the cinema... I remember seeing *Battlestar Galactica* in Sensurround at a Southampton cinema. I was so impressed with the way my guts vibrated to the roar of *Galactica*'s mighty engines that I went back the following weekend and saw it again.

My dad worked "behind the wire", as many people in our small town did – at the huge, sprawling compound of the Atomic Weapons Research Establishment at the edge of town. It was never entirely quiet in our town even in the dead of night - the AWRE emitted a steady background industrial rumble at all times, and you

could lie in bed at night and listen to the ticking of the criticality alarms guarding the reactors.

Having a parent working behind the wire brought certain privileges – access to sports facilities at one end of the compound, which was built on the site of an old WWII airstrip (I think everyone in our town at that time took their first unofficial driving lessons on the remaining bits of runway). Young kiddies were treated to an annual Atomic Christmas Party where they got to meet Santa and were given presents, and for us older ones there was the Rec. Soc. (short for ‘Recreational Society’) where they had a small cinema and, best of all, a subsidised bar where you could get cheap beer (government-sponsored underage drinking – class!) and where there would usually be a couple of coin-ops deployed.

One day, my brother came home from a session at the Rec. Soc. talking most enthusiastically about a new game they'd got in: “Like Space Invaders but there are all these little flies, and they peel off from their squadron and swoop down on you!”. Of course, I had to get on my bike and head over to the Rec. Soc. immediately to check out this novelty...

There, I found a lovely tabletop Galaxian machine (oh, how I miss tabletop videogames - they were so lovely, and just right for sitting with a mate supping a pint and having a nice two-player game) and I remember how impressed I was with it. Colour was rarely seen in videogames at that time, and even when it was it was usually via the Space Invaders method (strips of coloured plastic overlaying a monochrome display – hardly hi-tech stuff). But in Galaxian, colour was encoded into every pixel – bright yellows and reds and blues and purples – and the action took place against a smoothly scrolling

backdrop of twinkling multi-coloured stars. Astounding stuff for us PET hackers used to the limitations of a monochrome display and a fixed set of ‘graphics characters’.

The gameplay was sweet and much more flowing and intricate than the monolithic march of the Space Invaders. The brightly coloured, almost jewel-like insects hung in a buzzing swarm at the top of the screen, and just as my brother had described, individuals and groups would detach from the swarm with long looping trajectories, emitting strange warbling noises and raining sprays of bombs down on your ship at the bottom. They weren’t above taking the kamikaze option and ramming straight into an unwary player, either.

Gone was the saucer and shot-counting, and instead there were bright yellow ‘squadron leaders’ – destroy one of these when he was attacking and you would earn extra bonus points and a brief respite from the swooping attacks and rain of bombs. Pick off his wingmen first and then nail him for a nice fat 800-point bonus.

I spent many an afternoon happily hunched over that machine, either alone or opposite my brother or the Baughurst Piano Wizard, refining my skills, learning the little quirks of the game, tricks like destroying an enemy without actually firing your shot – simply touching the attacker with the nose of the ship. The cadence of the attacks was similar to Space Invaders, with the end of each wave being a frantic struggle to survive as the remaining enemies left the fleet entirely and continually swooped down the screen until finally you picked off the last one and there was a glorious moment of silence and completion before the next rack of Galaxians appeared, emitting their droning thrum.

Sound and colour... Smooth motion unconstrained by an underlying character matrix... These were things us PET geeks could only dream of, and seeing games like Galaxian put us firmly in our place. We could make little games for fun, sure, but they could never equal the wonders of proper coin-op games. And besides, I was running into more problems with my games. Again.

Part 3: Welcome To The Machine

The biggest problem, as ever, was speed. Certainly, things were much more efficient now that we could handle screen memory directly, but for some things the machine simply wasn't fast enough. Moving objects more than a single character in size was a chore using POKEs, and you couldn't do it fast enough to achieve smooth animation. You could go some way towards rectifying that by using strings of graphic characters interspersed with cursor-positioning characters - the old way was still the best when it came to big objects. Even that was slow though, and my games were starting to get too ambitious for the poor PET. I wrote a Battlestar Galactica game complete with animations of the Viper landing in the mighty landing bays and the Cylons swooping in on their

attack runs, and although it worked, the lack of speed was painfully evident.

With my games getting more complex and ambitious, they were also getting considerably larger, and beginning to tax the capacity of the PET's 8K of memory. It even started to cut in to my allocated daily computer time – the PET loaded and saved programs onto cassette tape, and it didn't do so quickly. Precious minutes had to be spent loading code into the machine at the start of your time, and saving it back again after. Many times I would arrive at morning registration or some class, breathless and late because I had been waiting for my precious SAVE to complete.

I had one game I was working on that was pushing the limits of memory - it may well have been the Battlestar Galactica game, that was pretty large - and one morning I came in to college to find the other geeks in a happy mood. They told me that the college had had the 8K PET upgraded to 32K of memory. I was delighted and impressed - in those days a 32K memory upgrade would have cost you about £500, and the ones I had seen came in external boxes almost the same size as the main PET system unit itself. The guys told me that our PET had had an internal memory upgrade, and so now we would have loads of space for our programs. And, indeed, when I typed the BASIC command to query the amount of free memory...

?FRE(0)

...the machine responded:

31740

thus indicating that there was indeed nearly 32K of free memory in there. Filled with enthusiasm, I pulled out the tape with my big game on it, loaded it up and began to code with gusto, adding whole new modules that I hadn't thought feasible inside the original 8K of system RAM.

After a while, something strange started to happen. As I was typing in lines of code, garbage characters started appearing at the bottom of the screen. The more I typed, the more random garbage started to appear. I was puzzled, and was on the verge of going to find a teacher to report that I thought the new memory expansion might be faulty - when the guys let me in on the joke. Someone had found a couple of locations in zero-page memory from which BASIC read its information about how much memory the system had. When you first turned the machine on, it would self-test the memory and place a value here indicating the memory size. When you typed the BASIC FRE(0) command to ask how much memory was left, BASIC looked in this location for the maximum memory size, and used that to calculate the answer.

However, that value was only set when the machine was first turned on, and there was nothing to stop a mischievous hacker - yes, I'm looking at you, Mr. Ruptured Rawlinson - from using the POKE command to change that value afterwards, making both the PET and some credulous fool - namely me - think that there was more memory than there really was. That garbage appearing on the bottom of the

screen had been the poor PET overflowing its memory into screen RAM as I'd gone merrily on my way adding great wedges of code.

I'd been had.

We laughed about that one all the way down to Grigg's pie shop. Me a bit more sheepishly than the rest of them. The bastards...

It was Rup, however, who showed me an entirely new direction to take – one that would cure all of my speed problems for the foreseeable future, and which would finally lead to my understanding exactly how the machine worked.

He told me what the SYS command did.

SYS was a command that had always been a complete mystery to me. It consisted of the word SYS followed by a number, and its action was unpredictable at best. On the PET it would often cause the machine to hang up completely, requiring a power cycle to start it again. On the TRS-80 it usually printed random strings of dots and asterisks. I had no clue what it was for.

Rup explained that the number was a memory address, just like for PEEK and POKE, but instead of reading or writing a number there, the PET would start executing something called "machine language" at that location. Machine language underlay everything on the PET - BASIC itself, which was there when the machine was turned on and seemed an intrinsic part of it, was just a program written in machine language. Each machine had a machine language type

determined by the type of CPU inside - our PET had a 6502 chip, so it used 6502 machine language.

The thing that really interested me about machine language was that it was supposed to be fast. Very fast indeed.

It was hard to find out much about it at first - at that time there would have been no question of anyone studying machine language at sixth form (this was an establishment still teaching CESIL, remember) so the library came up empty. In the end, to get me started, Rup lent me a photocopy of a manual for the 6800 chip – different from the one in our PET and so nothing I could actually write any code with, but enough for me to start to gain an understanding of how machine language worked.

It took a while poring over the 6800 manual in the college library – looking at example code and following through the steps of the programs by hand to see what was happening and how – before I began to understand... Machine language was both extraordinarily complex (in that to do the simplest things that in BASIC took one command - like `PRINT "HELLO, WORLD!"` - took hundreds of commands in machine language) and elegantly simple (in that there were only a few basic types of command and they all did simple, readily understandable things).

The insight I'd had when working directly in screen memory on the PET - that everything simplified down to being able to look at numbers and move them around - proved to be absolutely general, with a couple of extensions. The CPU chip was just a special kind of fairly simple calculator, and the memory of the computer was just a big

collection of numbers - 65,536 numbers was the most an 8-bit chip like the 6502 could 'see'. Those numbers could represent anything - letters on a screen, variables in a calculation... and now, I understood, commands to the CPU itself - 'op codes', they were called, in machine language.

The CPU could look at any of the numbers in any of the locations of memory. It could read them in to 'registers' - like the memory on a calculator - and do arithmetic on them. Furthermore, the arithmetic it could do was far more simple than even the most basic four-function handjob that we were used to using in college. The poor thing couldn't even multiply and divide, and it couldn't handle numbers bigger than 256, and it could only handle whole numbers - no fractions. There were other functions it could do - simple logic ops whose usefulness I had yet to fully understand - only using a CPU myself would teach me those aspects of it. But the basics were really simple. What the CPU did was to examine numbers, modify them in a few simple ways if required, and maybe move them around.

From those simple building blocks grew... well, everything. BASIC, biorhythms, games, databases, graphics on screens, Space Invaders, word processors, mailing lists in the Reader's Digest database... it was all just numbers, and computers, fundamentally, just took those numbers and moved them around, possibly molesting them slightly on the way in a manner which even the most mathematically challenged person could understand.

It was elegant and simple, and it was how all computers worked. To me, it was such a fundamental insight that I could not

imagine why it wasn't being taught on every comp.sci course in the land. Understand that, and it was like being able to see the Matrix – no matter how complex the output or behaviour of the machine, deep down, you knew what was actually happening.

It holds true to this day - sure, CPUs have got faster and a lot more complex, and they can 'see' a lot more than 65,536 numbers, can multiply and divide and even handle fractional numbers - but fundamentally it's the same. The windows you see on your screen, the porn on your hard disk, the mp3s you downloaded last night... all of them are just numbers, and your gigahertz CPUs are just looking at them and moving them around, occasionally gently molesting them on the way. They're just doing it very, very quickly.

The word 'digital' is bandied around routinely these days, and to most people it's simply marketing-speak, used to indicate a clearer sound or a sharper picture. Back then it wasn't widely known at all, and this simple, pure understanding was a revelation to me. I had finally seen the wiring under the board, the atoms from which the forthcoming 'digital revolution' would be made.

I knew how it all worked.

But there was still the little problem of actually getting it to work on our Commodore PET. I'd studied the machine language of an entirely different chip, and although the same basic principles ought to apply across any CPU, I needed to know the exact command set of our

machine's CPU if I was going to do anything other than crash it, nastily.

Our college library was devoid of anything specific about the PET, but I found out that one of the teachers – a chap we called The Klute on account of a superficial resemblance to a chess-playing alien out of Blake's 7 – had a copy of the hallowed Rodnay Zaks programming manual for the 6502, which was the CPU chip of our PET. A degree of Klute-hassling eventually persuaded him to cough up the sacred tome for a weekend loanage, with strict instructions that it must be back in his hands Monday morning. I took it home and did a spot more poring – most importantly, copying out the table which showed which numbers meant what commands to the CPU.

The PET itself wasn't set up natively to be kind to anyone wanting to program it in machine language. There was no 'assembler' - a program that allows you to enter machine code programs by using more human-friendly strings called 'mnemonics' to represent the commands. In an assembler you could type LD A #8, for example, and the assembler would know you meant a 'load accumulator' command with the immediate data of value 8 and generate the right code. Without an assembler you had to use your opcode table and work out that the machine language command to load the accumulator was represented by the number 169 followed by 8 for the immediate value.

Because BASIC was always there, you couldn't just randomly place your machine code program anywhere. BASIC was likely to use bits of memory you might have code or data stored in without asking politely, so where to put the code was an issue in itself. Fortunately

there was a tiny scrap of memory called the ‘cassette buffer’, where information was stored during the SAVE and LOAD operations that used the tape deck. At all other times this piece of RAM was unused, and so it became a good testing ground for my first faltering steps into the world of machine language.

Naturally enough, I wanted to begin with something simple, and I was reminded of the little BASIC program I'd written to prove the existence of screen RAM. That simple loop had the advantage of being extremely easy to implement in machine language, and furthermore would give a good indication of how much faster machine language was than BASIC, which we could plainly see took three or four seconds to complete the task.

During the rest of my lessons at that time I appeared to be particularly diligent, meticulously making notes on my pad during classes. But had anyone looked closer they would have seen that I was writing things like ‘A9 00 AA 9D 00 80 E8 D0 FB’ and drawing odd little diagrams. In an English Lit class.

Eventually, I was ready to try out my program. I had written it down in biro on a bit of paper, and an odd-looking program it was. It consisted of nothing but a list of seemingly random numbers, a loop which read those numbers and POKED them into the memory at the start of the cassette buffer, and a single SYS command which would hopefully cause the CPU to begin interpreting those numbers as commands and have it do my bidding.

One lunchtime I sat down, entered my strange program, and then saved it (so I wouldn't have to type it in again if I crashed the machine, which seemed, under the circumstances, highly likely).

I typed RUN – and was completely blown away. It was a pure jaw-on-the-floor moment. As I've already mentioned, the same program in BASIC took a few seconds to run - you could see the characters appearing one after another in sequence as they were POKEd into screen memory. But when I typed RUN now, all of the characters appeared on the screen at once. Instantly. They were just THERE. The same program that the PET took several lazy seconds to complete in BASIC ran in less time than you could measure in machine language. It was as good as instantaneous. Machine language had to be at least a thousand times faster than BASIC.

To confirm that was indeed true, I made my little program into an endless loop and had it draw the characters in white-on-black one pass and black-on-white the next. Running it yielded a flashing blur of characters at the top of the screen as the entire character set was redrawn again and again... far faster than the eye could see.

A thousand times faster than BASIC? Now here was a power worth the study to master...

I got a part-time job a couple of nights a week, cleaning offices and toilets at Lansing Bagnell, a big forklift truck factory on the edge of town. On work nights I'd take a packed tea with me and spend a happy enough couple of hours Hoovering round an empty office, with my can't-afford-a-Walkman-yet combination of cheapo mono tape deck

and some Sony MDR-5 headphones slung about me, listening to the Pleasure Principle by our sainted Gazza...

I could get round the office and do what I needed to do with a bit of time to spare, so I used to spend the last twenty minutes or so sat in the directorial-looking chair in the poshest of the offices on that floor, with my feet on the desk, eating my packed tea. One time, I was emptying the bins in the offices and found an entire book thrown away in one of the bins. Naturally I fished this out and had a lookie - it turned out to be a book of naughty stories by an author called Anaïs Nin. I had a bit of a snigger thinking about the bloke who worked in that office having a quick read of some naughty literature under the desk during office hours, and wondered why he'd thrown it away.

Too good for the bin, I decided, and filed it away for 'future reference'. Upon subsequent perusal I found that the stories were indeed quite naughty, but actually not badly written. One of them made fleeting mention of an animal called a vicuna (in a context upon which I shall not elaborate) and, intrigued because I'd never heard of such a beast before, I looked it up in a reference book and discovered it was a member of the South American camelids – vicuna, alpaca, guanaco and llama, all unusual and striking-looking beasts.

I had no idea that a chance find of a book of dirty stories in an office bin would indirectly and profoundly influence a large upcoming chunk of my life...

The after-college job provided me with a small income, and I was determined that I wanted to have at the very least some kind of videogaming equipment at home. Us geeks had found a new haunt to

bunk off to in town – a place called ‘Videotime’, which sold small home computers, videogames, and digital watches (this being so primitive an era that not only did people think that digital watches were a pretty neat idea, but also there were specialist shops devoted to the selling of them).

We’d gather at Videotime to lust over the likes of the Ohio Superboard, a ‘single board’ computer which consisted of one big circuit board without a case which hooked up to a TV to display the output. It had BASIC and graphics characters like the PET and was considerably cheaper, but still way out of my price range, after-hours job notwithstanding.

So I concerned myself with the videogame consoles. The Rolls-Royce of these was undoubtedly the eminently desirable Atari VCS. We became regular visitors to the shop and wangled many a session ‘evaluating’ the VCS by playing ‘Combat’, trundling our little tanks round the screen and blasting one another to oblivion. Once, I remember being in there when the very latest game came in – ‘Superman’ - where you had to fly the Man Of Steel round a multi-screen, Lego-like Metropolis, catching criminals, avoiding Kryptonite satellites and, bizarrely, Lois Lane who would, if inadvertently caught mid-flight, hang from your neck standing on one leg and chiming (she was supposed to be kissing you) and preventing you from picking up anything else until you put her down. Cue mumbled curses and mutterings of: "Gerrof, Lois!".

But even the VCS was beyond my budget, then. Although still relatively new on the market, and being by far the most advanced console, and hailing from coin-op legends Atari, it wasn't cheap (and although I couldn't afford one, I carried with me at all times a copy of

the game library catalogue – an early example of geek porn). In the end I settled for something a little less advanced - a machine called the ‘Teleng/Rowtron Television Computer System’.

This was a big, angular, grey-and-black box with analogue joysticks which had tacky button-pads underneath the sticks - the keypads were an influencing factor in my decision to purchase, because the catalog of ‘upcoming titles’ included a ‘Home Programmer Module’ which would supposedly allow you to hack the machine in machine code using the keypads to enter your code. Given what I was currently learning, the prospect of being able to actually code a dedicated game console was hugely appealing. Unfortunately the module never appeared, and it was to be many years before I'd get my hands on the innards of a console.

Obscurity notwithstanding, it was the first gaming system I ever owned, and it promptly took up residence beside the family telly. Over the next few months as finances allowed I acquired a small library of game cartridges – none of them as good as the Atari games I coveted but all pretty good fun. There was a reasonable clone of Combat, a bat-and-ball cartridge with a lot of variants on Pong and Breakout, a maze game, a pinball game with giant flippers made of huge pixels that looked like crinkle-cut chips...

Pretty soon, the flimsy handles on the joysticks broke off and I had to play by manipulating the little metal nubbin that remained. Still, we had some fun with that machine, and it was to last me for a couple of years.

But I still had the horn for Atari...

Part 4: The Joy Of Hex

Back at college, access to machine language was the final step needed for us to begin creating some proper games at last. Soon, the cassette buffer became too confined for serious programs, and we worked out how to force the PET to leave a section of main memory free for us to use for code, and how to make the cassette load and save routines work with data that wasn't lines of BASIC code. And I learned to count again.

It turned out that because of the way data is organised inside computers, CPUs like things arranged in clumps of powers of two, and so quite often you needed to refer to things in clumps of twos or fours or eights or sixteens.

Counting stuff that needs to be laid out that way is a bit awkward using human-familiar base-10 counting. If you choose a

different number base (in this case base 16 - 16 being a power of 2) then inspecting a computer's memory, and dealing with things like the op-codes of a machine language, actually becomes a lot easier.

Base 16 is called 'hexadecimal', and all it means is that you add in a few extra digits between 9 and 10. It's natural enough to use letters of the alphabet for these extra digits, so you count '0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,10'. When you reach 9F, instead of the next number being 100, you count 'A0, A1, A2'... and so on, until you reach 'FF', and only then do you get to 100.

It may not seem obvious HOW that makes things easier, so here are a couple of examples...

A computer holds data arranged in 'bytes', each of which contains eight 'bits' of binary data. The number of discrete states that can be held in 8 bits - and therefore the maximum number of numbers that one 8-bit byte can hold - is given by 2 to the power 8, or 256. The number '256' isn't a particularly pleasing or round one in decimal, but in hex it's a nice round 100.

Likewise the number of bytes addressable by an 8-bit CPU like the 6502 in the PET is 65,536. Again, an awkward number in decimal, but in hex a nice round 10000. So, if you're inspecting computer memory, a 10x10 table of memory contents printed in decimal would look confusing indeed, but a 16x16 table printed in hexadecimal would look much less so, and make it easier to find numbers based on the kind of 4, 8 and 16-number boundaries that often occur inside the machine.

You could actually 'read' machine language directly fairly easily if you used hex. To the CPU, any number in a byte could be a

command to do something - and since a byte holds 256 numbers, that meant that potentially there were 256 instructions the machine could understand.

Consider the 'LD A' command I used as an example earlier - in decimal that's instruction 169, so if you wanted to look it up in the list of instructions you'd have to count through to the 169th entry, which is a bit awkward.

But if you have a table of op-codes laid out in a 16x16 grid, and you label the rows and columns from 0 to F in hexadecimal, and look at the op-code itself in hexadecimal, you can quickly see it's instruction 'A9'. You just look in row A, column 9 and there's your instruction - easy.

Really, it is. Take my word for it. Honestly.

So, I wrote a little program (a 'monitor') to let me enter numbers into memory in hexadecimal, and to look at memory in hexadecimal - something that other machines had built in. This made entering and testing machine language programs much easier than all that mucking about with BASIC and POKEing values into memory by hand.

At first we used machine language sparingly - just little routines called from a still predominantly BASIC program to assist with the things that BASIC was slow to do, like drawing large blocks of graphics data. This allowed for bigger objects onscreen in our games. These hybrid programs were an ideal way for us to get used to harnessing the power of the CPU directly. Programming in machine language was a lot more painstaking than writing code in BASIC, and

it would be a while before any of us would attempt an entire game coded that way.

There was a degree of life outside college. My seventeenth birthday rolled around and I celebrated it by going hang-gliding near Marlborough. Gary Numan, the God of the geeks, came around touring the ‘Telekon’ album, and I went to the first music concert of my life, seeing him at the Mayflower in Southampton.

All of us geeks were major sci-fi nuts, and so the release of ‘The Empire Strikes Back’ warranted an expedition up to London to see it at the cinema in Leicester Square. It was excellent - I thought the Taun-Tauns were cute, and was quite sad when Luke had to slice one of them open.

Best of all, we got to spend time afterwards hanging out in the numerous arcades in that part of London. There was one that was a particular favourite of mine – the Piccadilly Arcade (now long since demolished and its location buried underneath the Trocadero). The Piccadilly Arcade was a proper arcade – the kind your parents warned you about. Dingy and filled with flashing machines and dodgy characters (indeed I remember hearing on the news once that someone got stabbed in there). But there was nothing seedy in my interest. It was simply the fact that they had lots and lots of good games. I played Targ and Galaxian and Space Invaders and sit-down Turbo... But there was one game in there which got most of my attention.

It was a big hulking sit-down cabinet with a large colour monitor inside. As I walked past, I saw the game spell out its name in

bright, glowing letters (and in blatant plagiarisation of the ‘Star Wars’ logo)...

“Exidy EXCITEMENT Brings You... STAR FIRE”...

...shortly followed by the attract mode, which revealed a first-person space shooting game heavily based on Star Wars – right down to the enemy craft having the same ball-with-wings shape as the TIE Fighters. This was of course irresistible to someone fresh out of the cinema from just seeing Empire Strikes Back, and I climbed eagerly into the cockpit with a Luke Skywalkerish attitude and prepared to spend a few happy minutes sending Darth and his chums off to oblivion.

The game was primitive by modern standards, but impressive for its time, in which first-person games of any description were rare indeed. For control you had a throttle lever and a flight yoke, and before long you found yourself swooping through the simple 3D starfields with ease, chasing squadrons of TIE fighters, dodging shots, getting a ‘laser lock’ on the enemy ships and seeing them break apart into colourful chunks as your lasers hit home.

Every now and then, a differently-shaped ship - looking a lot like a Colonial Viper from out of Battlestar Galactica with ‘EXIDY’ written on the side - would fly by, and if you managed to line up and shoot that ship you would get a bonus and the words ‘GOT US!’ would appear in large, flashing letters.

Simple though it was, Star Fire was one of the first ever games that I found to be truly immersive. Sitting in that cockpit with that large colour screen filling my field of view, it felt for a while like looking out of a craft instead of merely remote-piloting one from above, as was the norm in most other games. I became pretty good at the game, and before long could post up a boardful of 'JCM' hi-scores which would still be there on returning to the arcade a few hours later – no mean feat for a machine in a popular arcade in London.

In college, some of us would occasionally attempt versions of popular arcade games, in our own simple way on the PET. Rup put together a pretty decent Space Invaders, and I decided that, as my Big Project, I would attempt to bring something of Star Fire over to the PET... and that I would write the entire program in machine language.

One problem we had with the PET was that there was no sound generator built in to the machine, which made designing satisfying games a bit harder - games aren't really games without a few electronic bumbles to signify the firing of shots and the explosions of enemies.

However, we discovered a way to work around it. If you placed an AM radio near the machine while it was working, the CPU would generate RF noise that could be picked up as interference on the radio. And we found that if you altered what the CPU was doing you could cause variations in that noise. If you stuck it into a tight loop, and then caused that loop to lengthen slightly with each iteration, then you could actually hear quite a passable 'zap' noise in the interference on the radio.

And so that's how the sound effects were generated for my first ever all-machine-code game -from a radio standing on top of the cassette deck picking up RF interference tuned into sound effects by sculpting CPU loop lengths!

Eventually the teacher whose name I forget but who looked a bit like Phineas out of the Freak Brothers came up with a small hardware device which attached to the I/O port on the PET and which could generate sound by toggling one of the I/O bits, and that made for slightly better sound effects, but I still have a special fondness for our AM radio hack.

Star Fire was to be my PET magnum opus, taking several weeks to complete. Hand-compiling the machine code for a game of any reasonable complexity was a painstaking task. But between us, me and the rest of the PET geeks had amassed a fairly large body of work by this time, and we were never short of games to play when we weren't directly engaged in creating more.

In the second year I was at sixth form, the computing department had been moved into its own small building – a wooden hut with a few small rooms, each of which housed one machine, and one large classroom. People - even some non-geeks - would congregate in the PET room to play the games, and, since this was before it was really practical to save hi-scores, before long the walls of the PET room got covered in hi-score lists as people wrote down their personal bests on the wall in biro. This eventually came to the attention of the Powers That Be, who were not amused and the whole lot had to be painted over.

We also acquired a new PET - still only 8K, but with a green screen and a proper keyboard instead of the somewhat cramped calculator-style one on the original. On the very day it was delivered I was queueing up to get a crack at it. I managed to be the first one on the new machine, and I duly loaded up my game.

Steering in Star Fire was accomplished using the numeric pad - 8 for down, 2 for up, other directions and diagonals derived from the keys around the '5' key – and I think it was 'space' to fire. In the course of a game, the keying action could get quite hectic as you scrolled the field of little PET-character TIE fighters around into your laser sights. I'd even put in an Exidy-style bonus ship that would come around once in a while and which, if shot, would refuel your ship and restore your shields (in the wonderfully weird logic of games of that era).

I sat down at the new PET and began to play - noting with admiration that there was a lot less 'snow' interference from accessing video RAM than on the old machines (this was way before I'd heard of VSYNC).

I got into the rhythm of the game and wave after wave of green TIE fighters succumbed to my lasers. Then, in the throes of manoeuvring my ship with the numeric pad, my finger slipped off a key that I was holding down - and with a SPANG!, the key sprang up and flew off the keyboard and halfway across the computer room - just as Phineas Freak came in to check that the new machine was working properly.

Fortunately, it proved easy enough to clip the key back onto the keyboard again, and I escaped with a minor bollocking, but for a

few moments I'd been convinced that I'd broken the new machine within minutes of its arrival.

One night after college, the parents were out and I was lying on the couch idly watching the news, when a story came on which made me sit up and take notice.

A man called Sinclair – a bit of a boffin from the looks of him – had designed a really cheap computer, and he'd gone to a school with some prototype machines for the pupils to try out. The news footage showed pupils sitting in front of televisions to which were attached odd-looking little machines, looking a bit like squashed calculators but without any LED numbers.

The keyboard appeared to be printed on, being totally flat, and - at the time I assumed it was some kind of TV interference - the display on the tellies seemed to flicker whenever anyone entered a keystroke. It was certainly no PET - but it was recognisably a computer, evidently running BASIC - and it was to be introduced at a then stunningly low price of £100. A price that even I might be able to afford, given a bit of saving and cleaning of the offices at Lansing Bagnall.

When my parents came home, I excitedly told them about the cheap computer I'd seen on telly, and although at first they were dubious (why did I want one of those? surely I'd get one and then it'd be unused in a cupboard after a few weeks...) but, eventually they relented, and I even extracted a promise that if I could save up fifty quid on my own they'd match that so I could have the machine sooner.

Excellent stuff. At last, nearly two years after I'd first set eyes on a computer, there was a real chance that I could get one of my own. The idea of having your own machine that you could use exclusively and never have to worry about the next person shoving you off to claim their time slot was tremendously exciting. I don't think I'd keenly anticipated a purchase so much in my life. I went on cleaning my offices at Lansing Bagnall and putting away a few quid each week towards my goal.

A couple of months after the news piece on telly, the little machine was released and adverts and articles about it started appearing in computer magazines. Us geeks devoured these hungrily, soaking up every detail and collectively drooling in anticipation. The machine was called the Sinclair ZX-80, and it really was a barely minimal machine - made out of snap-together vacuum-formed plastic, with a small membrane keyboard, and with only 1K of memory inside (you could buy a plug-in 'RAM pack' to increase the memory to 4K). But still, we were ingenious coders, and I was sure that somehow I'd be able to make some simple games even in that tiny amount of RAM.

Demand for the machine was overwhelming the small company that Sinclair had formed to sell it, and at first there was a six-week waiting time between ordering one and actually receiving it. I couldn't afford to order mine yet, but Rup managed to acquire some funds and he ordered his straight away. Eventually, it arrived, and the same weekend I set out on my bicycle to visit his house so that I could finally have a hands-on look at it.

We sat and geeked the evening away, writing small BASIC programs on the ZX-80's odd little keyboard. It was strange. It insisted on entering whole keywords with a single keypress, and there were different SHIFT modes you had to use to get particular keywords, and some you had to type in full (these were listed in a small sticker on the front of the machine's white plastic lid). Not at all like the PET's screen-oriented editing, and a bit awkward at first.

And that flicker I'd seen on the news piece wasn't TV interference at all, it was a result of the fact that if the little machine was actually doing ANYTHING else apart from generating the display - running a program or even reading a key-press - then it had to turn off the TV display. This was a little disappointing. How were we going to make games move if the computer could only run programs with the screen off? It also made entering code a somewhat epilepsy-inducing experience.

None of which was enough to put me off buying one. Just the prospect of being able to write code, at home, on my own machine, whenever I felt like it was enough for me to forgive any of its little foibles (and it had plenty of 'em). I continued saving my Lansing Bagnall money...

Back in the real world, I carried on doing what I'd actually come to college to do - get my A-levels and ultimately a place at University. I can't say I shone at any of my subjects, mainly because most of my learning and studying energy was focussed entirely on the school's computers.

By that time, me and Rup knew considerably more than any of the teaching staff about the minutiae of how CPUs worked just through the practical knowledge we gained from writing our games. But, unlike Rup, who was properly on the comp.sci A-level, I would end up with no qualification to show for it, since I wasn't on the course and wouldn't get to take the exam. I muddled along, barely competently, in maths, physics and Eng Lit.

I got conditional offers from a couple of Universities and was invited along for interview. The first place I went to was the University of Aberystwyth - a Welsh seaside town I'd visited when I was a child going on holiday to Wales every summer. I took the train there and arranged to stay overnight in a bed and breakfast - quite an exciting thing for me in those days, since I hadn't been away from home much.

I remember being met at the university by a girl called Sian Rhys, the daughter of a Welsh family who'd been our neighbours when I was a kid. She was already a couple of years into her degree at Aber, and she kindly met me and showed me around, and told me about the vast number of pubs in Aber, which sounded OK to me.

About the interview itself I don't remember a lot - I am certain that I enquired about the possibility of transferring to a computing course, but was told that the A-levels I was going for would admit me only to the physics course I'd specified, and that comp.sci courses were oversubscribed, and I needed the matching A-level to get a place on one. Apart from that, it was standard stuff. I'd get in if I got certain grades on my A-levels.

Interview over, I had the evening to myself, so I set out in search of sustenance, and had fish and chips and, in one of the many

pubs, a couple of beers. As it was a seaside town, there was a pier, and at the foot of the pier was an arcade. Replete and happy after my couple of libations, I set out in search of gaming...

I don't remember what else was in that arcade, because one machine captured my full attention - a big black Atari Lunar Lander vector cab. The objective of the game was simple - control the orientation of your lunar module with the buttons, and fire the craft's engine with a big, satisfying throttle handle.

The sound on the cab was turned up very loud and the machine shook with the roar of the engine when at full thrust. You had a limited amount of fuel with which to decelerate your craft from orbit and achieve a soft landing on one of several plateaux amongst the lunar mountains, depicted in glowing white vectors.

The game was tricky at first, and my first few attempts ended with the module flying into pieces and a little message appearing telling me that I'd destroyed a very expensive spaceship and made a large crater in the ground. The controls were simple but gave good control once mastered, and before long I was able to set the spacecraft down intact – at first on the widest and lowest-scoring plateau, and eventually, gently easing it down onto the smaller, higher-scoring, but much trickier landing sites.

I really loved the feel of the game – bleeding off the speed with prudent firings of the rocket motor, taking advantage of freefall and inertia to get where I wanted to go without using too much fuel, but being certain to allow time and fuel to decelerate and land gently before the fuel ran out. It felt really good being able to interact with this

simple physical simulation, and it planted in my mind an appreciation of such games that I still have to this day, especially when I fire up STEem for a game of 'Virus' or VICE for a bash on 'Thrust'.

Eventually, the college year started to come to a close and it was time for the A-level exams. I really can't remember that much about the taking of them - it was the usual curious exam-time mix of anxiety and freedom, in that you had to actually sit the exams, but in between them you had no lessons and were free to do whatever you like. In our case, this meant driving round our usual haunts in Mole's Vauxhall Viva, or just mooching round Basingstoke (and, of course, playing our games on the college machines).

Then, it was all over. I remember that we all had leaving interviews with the head teacher (and I got bollocked one last time for not wearing a tie) and then we were free – out of the educational system for the first time in our lives. Not yet sure if we'd get to re-enter it as University students after the summer.

And finally I got enough money together and put in my order to Sinclair Research for my very own ZX-80.

A few weeks later, one morning, two items of mail arrived addressed to me. My mum brought them into my bedroom, along with a nice cup of tea. One was an envelope containing my A-level results, and the other was a parcel from Sinclair Research, containing my ZX-80.

The contents of one of those items was to determine the path of the next twenty-four years of my life...

Part 5: Fur-ther Education

I didn't bother opening my exam results straight away. I'd agreed with a mate of mine who was getting his on the same day that we'd not look at them until we were safely down the pub with a pint in front of us. Instead, I opened the brown parcel from Sinclair Research and there, nestling in the polystyrene, was the little ZX-80 in all its vacuum-formed, membrane-keyboard glory.

I plugged it in to the family telly and fiddled with the tuning, until eventually I saw what was to become a familiar sight - a white screen with a black inverse-video letter K in the bottom left-hand corner. I fiddled with it for a couple of hours (slightly irked at the screen twitch whenever you pressed a key, and the fact that when it was

actually running a program the display turned off - but hey, it was my own machine). Then it was time to meet my friend (the Baughurst Piano Wizard, also known as James Lisney, who in a few years' time would be involved musically with some of the Llamasoft games).

It was a nice sunny day, and we walked several miles to a country pub where we'd been underage drinking for some time, acquired a pint of Guinness each and went outside to sit in the beer garden and look at our results. There was a large Alsatian dog that belonged to the pub in the beer garden, and just as we were about to open the envelopes, he raised his tail and released what can only be described as a jet of projectile diarrhoea from out of his arse. It smelled really awful in the summer heat, and I hoped it didn't presage the arrival of shitty results.

I think the BPW failed one of his A-levels, but he really didn't care because he had a super A ultra-double-plus in music as expected (he was the Baughurst Piano Wizard after all), along with a guaranteed place at the Royal College of Music. My own results weren't brilliant, but nor were they catastrophic - I'd passed Eng Lit and Physics reasonably well and scraped through in Maths with Mechanics. It wasn't enough to get me into Aberystwyth Uni, but through the confusing vagaries of the admissions system, it was sufficient to get me a place on a physics course at East Anglia.

Secure in the knowledge that we weren't doomed to be ejected from the educational system that summer, we demolished another couple of pints of Guinness each before setting off home mildly inebriated. Jim had to play tennis that afternoon and I wondered whether his state of mild pissedness would interfere with his ability to play. No such exertion for me - I returned home and fell to playing with

my new toy, finding my way around the little beast by writing a few simple little programs.

That was basically the pattern of my summer holidays that year - playing with the ZX, poking around to find out what was in there. One of the reported quirks of the machine was that, as the basic model only had 1K of memory, it didn't have a memory-mapped screen like on the PET, where the screen alone took up almost a whole K. It had something called a "serial display file" instead, which meant that the old technique of POKEing values into screen memory didn't work.

I thought it was silly to say that the display wasn't memory-mapped. After all, the characters on the screen definitely were in memory somewhere; surely all one had to do was find out where. And so I set out to explore the tiny ZX-80 memory, eventually working out a trick - printing lines of known length to the screen and allowing for the fact that there would be an invisible delimiter character at the end of each line - which allowed me to create a display which could be addressed just the same as on the PET. You had to be careful what you POKEd to the screen, though - there were far fewer displayable characters than on the PET, and some characters caused strange things to happen if you tried to print them. Some warped the display in an alarming manner, and some caused the machine to crash completely.

I wrote a little program to set up this memory-mapped display and to demonstrate how it could be used, and, along with a letter explaining how it worked, sent it to one of the early computer magazines, 'Practical Computing', which had just started a page dedicated to us new ZX-80 owners. The next month when I walked to

the shop to pick up my copy I was delighted to find that my letter and program had been published on the ZX-80 page - my first taste of geeky fame.

Although people did pull off miracles of coding in that tiny 1K memory – amazingly, I remember somebody actually managed to create a chess program with rudimentary computer AI – it really wasn't enough, and I eventually ordered the 4K RAM pack. Just the RAM pack, mind - I couldn't afford any actual memory chips to go in it initially, so I bought it 'unpopulated', and then as funds allowed bought 2114 chips one at a time, expanding the system memory by half a K at each increment.

I got some information about the Z80 CPU that was at the heart of the ZX-80, and taught myself its machine language. Because memory was so tight in the machine, it was difficult to find a safe place to POKE some machine language where it wouldn't get stomped on by the system stack or the dynamic display file. I discovered the same trick that lots of us ZX machine language hackers came up with - if you made the first line of your code a REM statement, then BASIC wouldn't try and execute it (of course), and the text of the REM statement would begin at a static, known location. Therefore you could reserve x bytes of memory for machine language by simply creating a REM statement x characters long, and then POKEing the machine language commands into the statement from the known base.

You had to be careful, though. If your machine language routine contained the same byte that represented 'end of line' to the ZX-80 BASIC interpreter, you'd chop off the end of your REM

statement and leave garbage following it, which would often cause the machine to crash. And if your code contained any of the values that distorted or crashed the display, then when the BASIC interpreter tried to list the program, at best it could make the display unreadable, and at worst it could crash the machine outright, losing your work as it did. Still, we worked round these problems, and many a ZX-80 program consisted of a single REM statement full of gibberish followed by the line `RANDOMIZE USR 16514` – the BASIC command which transferred control to the machine language stored safely in the REM statement.

I tinkered away writing bits of code to do things that BASIC was slow at, just as on the PET - Conway's cellular automaton, screen scrolling, that kind of thing. I pored over disassemblies of the ZX-80's ROM code, eventually working out an inelegant hack - simply repeatedly calling the ROM routine to regenerate the display, and sneaking in my own code between calls - to allow the creation of graphics which moved without the display turning off.

Since I couldn't be taking over the family telly with all my hackery I looked in the free adverts section of the local paper and bought my first ever telly - a monster of an old black-and-white beast, just about barely alive, for a fiver from a granny in Reading, so that I could lurk in my bedroom and geek away to my heart's content.

Soon my summer of geeking came to an end and it was time for me to leave and go to UEA in Norwich. My parents loaded up the car with all my gear and my ZX-80 and my Teleng-Rowtron games

system and that clunky old telly and we set off. Unfortunately, halfway there the car broke down, and I was sent on by train with a subset of my possessions to find my own way to the University halls of residence. I found my room, upstairs in O Block, and settled in as best I could. And then I did what every newly arrived student doubtless did, everyone a stranger and not really knowing what else to do - I headed to the student union bar and got a bit pissed.

My main memories of that first night are being a bit shy and nervous, but eventually as drink reduced my inhibitions, I fell in with some guys also from O block. I remember that in the bar they had a Star Trek pinball and one of the really old, big Cinematronics Space Wars games, and predictably enough I spent a lot of the time hovering around that thing, mostly watching others playing and occasionally having a go myself.

I also started drinking far too much cheap student union cider. Turns out the blokes I'd hooked up with were, well, much more 'laddish' than me. I wasn't that experienced at drinking, in part due to a formative experience which I shall shortly relate. The sloping off for a couple of illicit pints with the BPW had been very much an occasional thing, rather than something done every night or even every weekend, so I just wasn't experienced at supping, and my tolerance was very low. In order to keep up with the lads from O Block, I was having to drink rather more than I was used to, and smoke fags too, something I didn't do at all at that time.

I drank lots at the bar, and then we moved on to a concert that was being held that night in the halls, a band called Fischer-Z, about whom I don't recall that much except that they were very loud and at

some stage in the proceedings someone came on stage dressed as a giant spliff.

More tins of lager were bought and consumed during this, and by the time I got back to my room I was thoroughly pissed. I flopped into bed and, unfortunately, rather than being claimed by merciful oblivion, I found myself experiencing a bad case of the helicopters. This swiftly deteriorated into a litterbin vomiting session (we didn't have our own loos, and there was no way I was running halfway down the block to the communal bogs with my stomach contents lapping at my back teeth).

Unfortunately, this process proved to be iterative, and I ended up spending most of the night trying to work my way through Piers Anthony's 'Ox' and 'Orn' stories (which I recall invoked Conway's cellular automaton at times as part of the story), occasionally pausing to deposit some sick into the bin. Eventually, as it started to get light again outside I went for a bit of a walk to try and make myself feel better, and found a nice calm little bit of woodlands, just down the road, where I rested a while, and vomited a bit more.

I had only been as pissed - and as badly hungover - once in my life before. When I was about 14, our school did a French exchange programme, and some of us went to stay with French families near a town called Alençon, which has the dubious honour of being twinned with Basingstoke. I'd stayed with a family called Beloeche, twinned up with their son Eric. Now Eric Beloeche was very mature for a 14-year-old, in that way that can seem a bit intimidating to a shy, thin and unremarkable British teenager like me.

Eric was a nice bloke though, and I spent some good time with him and his family over there, learning what it was like to go to French school and ride a bike on the wrong side of the road and how French people cooked a steak by just showing it to the cooker. And in due course he came over to stay with us, and his rugged Continental good looks and confident air made him most desirable to the girls in my year at school.

As a result of this, I got invited to a party I would never otherwise have even known about (being shy, thin and unremarkable) out in a village called Cliddesden way over on the other side of Basingstoke from where I lived. The party wasn't even with any of my small peer group in school, and I'd blatantly only been asked to come because I'd bring Eric with me. Perhaps I should have given it a miss, but matters were complicated by the fact that there was a girl going to the party; a girl upon whom I had a (completely innocent, utterly silly, shy adolescent kid style) bit of a crush. I got my parents to deliver me and Eric to Cliddesden village hall on the appointed evening.

Of course I had a very awkward time at first. These people weren't my mates, I was shy and awkward, and Eric was away with the girlies straight away, so I shambled around on my own trying not to feel too awkward about being far too inhibited to actually move in any way rhythmically to the little village hall-style disco they had going on.

Now this was a party for 14-year-olds, and there certainly shouldn't have been any alcoholic drinks there, but upon a foray to refill my glass of Coke I noticed that there were in fact several bottles of Strongbow around. At that point I don't think I'd ever been inebriated in my life - I'd probably had a sip of Mum's wine at Sunday lunch once or twice, but never enough to actually have any kind of perceptible

effect. Still, I was bored, and so I thought, if only for something to do, I'd help myself to a bit of Strongbow. I filled my plastic cup with the fizzy yellow liquid.

It didn't taste too bad - sweet but with a sharper tang to it, and after I'd finished my first cup I was starting to feel more relaxed and not quite so awkward. So I returned to where the cider was and refilled the cup and drank a bit more.

This, too, proved to be an iterative process.

The results, on the whole, were entirely predictable, and I like to think actually helpful, once I'd recovered from the projectile vomiting and the hell that was my virgin hangover the following day; for they taught me quite fiercely that a little of something can be good but a lot of it can be Very Bad Indeed.

I got very pissed, made a complete twat of myself in front of a lot of people from my year in school, actually tried to talk to the girl I had a crush on whilst completely off my face and barely able to speak, thereby ensuing that whenever I saw her in school I wished devoutly for the ground to open up and swallow me. I spent a lot of the time lying on the ground outside, vomiting, and indeed decorated the back seat of my mortified parents' car when they arrived to pick me up. I learned that one must treat alcohol with caution, and above all I learned to be completely embarrassed and ashamed.

You may wonder, and rightfully so, why I am dwelling upon matters of teenage drunkenness when I am supposed to be telling the

history of Llamasoft, and we'll get to that if you'll bear with me. Because somewhere on that awful slide down to drunken twattishness and shame, something happened; something insignificant at the time and for some years to come, but which I remember quite clearly.

Recursive flashback - forgive me, but it's a seed event...

The first music I ever really got into, as opposed to just heard as a background to my five older brothers growing up, was Pink Floyd's *Dark Side Of The Moon*. I was 11 when that album was released, and far too presentient to make any coherent music purchases of my own. My brother Steve bought a copy of *Dark Side*, and in due course I heard it and I liked it – in an entirely different way than I'd ever liked any music before. It was something I really wanted to listen to, again and again. I made a recording of it by putting a microphone in front of one of the speakers of the family stereo (playing in Mono mode) and recording it onto a cassette, and I'd take that crappy recording and lie in my bedroom with the lights off, just listening to it. At that age I wasn't really too into the singing bits (time would reveal their beauty to me). But the unique sound, the synthesisers, the... Floydianess of it really clicked with me.

In particular, one track, *On The Run*; a monster synth riff with aircraft noises and wicked filter sweeps. For some reason I started to “see” the music as progressions of geometric shapes and colours in my mind's eye, lying there in the dark. Only Floyd did that to me. It just seemed like a natural way for my imagination to accompany their music. Hence my predilection for listening to them in the dark.

Anyway, back to that night of 14-year-old-drunken shame... As I mentioned earlier, they had a small disco in Cliddesden village hall that night, and the DJ had some of those little 3-lamp traffic-light-style flashing lights you used to be able to get from Tandy, and somewhere along the trajectory down into irrevocable nonsentience, something must have swilled loose in my brain on the tide of cheap cider.

I thought about my Floyd-inspired internal visualisations and what those silly traffic-light flashers were actually trying to do, and I looked at them and thought to myself, quite clearly and distinctly... "There must be a better way of doing it than that"...

In the months that followed, despite being crippled with embarrassment for most of the time on account of how awfully I had behaved that night, I started to have strange daydreams during the more boring parts of English lessons. Most kids daydreams are all about being a footballer or a rock star. Mine were similar, in that there was music, but I wasn't playing it, I was doing... something... that involved... big screens of some sort upon which were... something... a bit like what I used to see when I was listening to Floyd, except I was "performing" it along with the music rather than just imagining it... somehow...

I had no idea how any of this would happen. I'd never heard of computer graphics and I was of an age where you just assume that if something is a good idea then it will get made, and all I'd have to do was find out where I could get such a machine and then learn how to use it. Just like buying a guitar, except... different.

(And somewhere in America, entirely unknown to me, people at Atari were smoking their board-meeting spliffs and coming up with the Atari Video Music... and John Whitney was experimenting with visual harmonics using PASCAL programs and a frame-by-frame film recorder)...

Back at University, that first night and following morning of puking reminded me of the lesson I'd learned back when I was 14, and indeed probably occurred for all the same reasons - feeling alone and a bit nervous and shy in a social situation in which I didn't know anybody. But again, it probably stood me in good stead in the long run, because I didn't end up being a very boozy student at Uni. A bit is nice, too much is unpleasant; avoid the extremes and you'll be fine. Not a bad lesson to learn in the end, and probably cheap at the price of two hangovers, and a semi-infinite amount of shame.

Part 6: Higher Learning

I'd like to say that my time at University was successful and productive, but if I did I'd be fibbing. The bottom line was that I'd got into Uni on the basis of a set of A-levels that were geared towards doing Physics, and Physics simply wasn't what I was interested in any more. Academically, I flailed around, found the maths really difficult, and generally wasn't motivated enough to knuckle down and put in the study and learn enough to keep myself afloat. And besides, I spent far too much time playing around with the computers.

Socially, I had a much more productive time, and after the initial couple of weeks hanging out with entirely the wrong crowd for reasons of needing to belong to some peer group, I went for a wander

one day and, walking through E-block, came across a door upon which someone had taped the logo from Pink Floyd's *The Wall* album. Rendered in dot-matrix printout, too. Pink Floyd and computers - I thought there was a fair chance that I might have something in common with whoever lived behind that door, so I knocked and introduced myself, and met Andy, who was to become a good friend in my life from that time onwards. As a result of that Floyd-inspired meeting I fell in with a much more suitable group of friends, and the need to get mortal drunk among strangers in order to fit in vanished completely.

Life at Uni was punctuated by, and anchored on, moments of gaming. We lived in halls a couple of miles off campus, and the day always started with a bike or bus ride. Lunchtimes were spent in the little games room in the Student Union hall, where there was an array of pinballs (*Black Knight*, *Gorgar*, and, memorably, *Nitro Ground Shaker* that some nutter got mad at and tore the head off) and videogames such as *Tail Gunner* – with its lovely vector graphics and metal analog joystick – and the sublime beauty of *Asteroids*. On the other side of the room, there was *Galaxian* (which you could sometimes get credits out of by flicking 2p up the coin return slot) and, of course *Space Invaders*.

I'd retire to the arcade at lunchtimes with my Healthy Student Lunch (usually a can of Coke and a packet of Jaffa Cakes), play or spectate as finances dictated, and hang out with the other gamers of that year's intake. Chief amongst those was Chico, who lived a couple of doors down from Andy of the Floyd printout, and who was another of our circle of mates. He was one of those people who are just awesome at videogames by default. I'd encounter more of that type years later at

Atari test, but Chico was the first I'd ever met, and he was simply awesome at any game he chose to try. You'd go into an arcade with him and he'd get on a new machine for the first time, never having seen it before, and effortlessly rack up a massive hi-score on his first go.

The complete bastard.

I'd thought I was good at Space Invaders until I met Chico. He was so good at Space Invaders that he could play it forever - it was no longer a challenge to him at all. So if he played the game at all he'd set himself some task that us mere mortals simply could not even begin to comprehend. Shooting all the Invaders was too simple. He'd time his shots so precisely that he could shoot individual Invaders out of the flock to eventually form his initials from the remaining attackers.

He taught me about the Execution Method, where he exploited a bug in the game that meant the Invaders couldn't bomb you if you let them come down to the very last line before they'd land and end the game. By shooting a 'window' in the Invaders' ranks, you could sit in complete safety for most of the game gorging yourself on 300-point saucers, but in order to progress you occasionally had to endure a nerve-wracking few moments of ultra-precise timing as you shot the last few Invaders moving directly over your shooter at ultra-low level. Miss one and the game would end instantly. It was a satisfyingly flashy trick to master, and elevated me to the ranks of not-too-shabby Space Invaders players. But I never got as awesomely good as Chico.

At anything.

The bastard.

There was more gaming to be had back in halls, where there was the Star Trek pinball and Space Wars as already mentioned, and a Future Spa pinball and another Space Invaders machine in K Block (which could occasionally be coerced into giving a free credit with judicious pressure of the knee on the coin door). It was in K Block that I reached a significant point of Space Invaders zen – I gained perfect knowledge of how the shot counting trick worked. I don't just mean that I could count the shots in my own game and always score 300-point saucers - that was just usual moderately skillful play. I mean that I could be anywhere in the room in K Block within earshot of the Invaders machine, talking to people, doing whatever else I might be doing, and if someone was playing the game, I could tell them instantly what shot number they were on. My brain automagically processed the sound effects from each game played and kept track of shot numbers without any conscious effort at all on my behalf.

There was gaming available in my room in O Block too, in the shape of my old Teleng/Rowtron console upon which matches of Gravity Tennis were played, and on my ZX80, which I spent many a night hunched over, bathed in the flickering monochrome cathode rays.

Being a student, in charge of my own finances at last, I decided to use them wisely, and figured that a term of eating baked beans was a small price to pay for finally assuaging my long-held lust for the Atari VCS. So I set out on my bike one day and rode to BB Adams at the end of the delightfully named Upper Goat Lane in

Norwich and bought myself a VCS and a copy of Space Invaders (112 game variations!). I rode back to halls somewhat precariously, as the VCS box was quite large and awkward to carry on the handlebars of my bike, and spent the first of many afternoons worshipping in front of the VCS. My popularity increased with this purchase, and my room in O Block became a regular venue for sessions of Combat, Space Invaders and Basketball (but not Golf, which was complete crap, and looked more like a robot whacking a pixel with a pipe made of Lego than actual golf).

Midway through my time at Uni, Uncle Clive introduced the ZX81, which was both cheaper and better than the ZX80. I couldn't afford to make the swap (blowing all my grant on a VCS hadn't helped much), but I did send off for the ZX81's ROM, which was a drop-in replacement for the one in the ZX80 and upgraded the machine to have nearly all the features of the ZX81 (apart, sadly, from the improved display that didn't flash when you entered a key-stroke).

But now I had a somewhat improved BASIC in which I could actually use floating-point numbers and rudimentary graphics commands, and I tinkered with that for a while, making plots of 3D surfaces that looked like they were made of black Lego on snow. A guy I knew in P Block had an Acorn Atom, a much nicer machine with a 6502 like in the Commodore PET and dot-addressable graphics. Some nights I'd go and hang out in his somewhat evil-smelling room and hack on his Atom, producing a range of free-running abstract graphics generation programs that we'd get stoned and sit and watch.

There were computers to be fiddled with at Uni too - PETs and TRS-80s over in the comp sci block, Apple IIs downstairs in Biology which were the cause of my buying my first ever floppy disk (not drive - I just mean a single 5 1/4-inch floppy) to store my code on. Then there were the University mainframes... a Vax machine to which we all had quite restrictive accounts which allowed only a small amount of time each day (nowhere near enough time for the geekily-inclined).

The Vax installation was relatively new, and the thing was full of holes. We found that by carefully pressing Ctrl-C at the right time, we could drop out of the restrictive shell that we were allowed to use, and into a level where we could do whatever we liked - look at all the files on the system, find anybody's password, edit our own account status and allocate ourselves as much time as we liked.

We wrote little scripts which made it look like we were using the intended shell while really allowing us access to the advanced stuff underneath. We never did any damage and basically had a fine time for a couple of months before we got rumbled and hauled up in front of the Dean of the computing facility, who ticked us off and told us that we could have been thrown out for what we did but actually they were rather pleased that we'd found some holes in it so they could fix them.

That was pretty much the limit of my nefarious hacking career.

So, life for me at Uni was basically a round of gaming, coding, hacking, hanging out with my mates and getting stoned and listening to Pink Floyd round at Andy's room in E Block. All of this

was rather good fun and excellent for my social and gaming life, but didn't really do much for me in terms of getting a degree in Physics. I became less and less competent and eventually failed my first year exams, and despite doing re-sits (which was a pretty pointless exercise, given that I did sod all to prepare for them) I was given the ceremonial order of the boot at the end of the first year.

I remember going to what they would call an "exit interview" these days, with my tutor from the Maths and Physics department, and pointed out to him that I could actually answer a lot of the questions on the Comp Sci degree paper, and if they'd only let me back onto a second-year computing course I'd be a happy little chappie and probably do rather well. I was told that it would be impossible to "bend the rules" to accommodate this scenario, and that I should get onto my bicycle. I mentioned out of the blue that I was even thinking of starting my own software company, and I was roundly told that the market was far too competitive, it'd never work, and that I should get a grip and find a place in a polytechnic the following year instead.

And so my Uni time ended - with a memorable night to round it off. Andy, being a member of the Pink Floyd Fan Club, managed to get tickets to see the very last night of The Wall live at Earls Court. I left a maths exam early to get to that gig, and I don't regret it, since I hadn't a chance of passing anyway.

Eventually, I ended up back at my parents' place, in disgrace for having 'failed', but with my VCS and upgraded ZX80 and a summer of doing bugger-all ahead of me. On the day I arrived home,

my parents were out, so I installed myself and VCS in front of the family telly (where I finally got to see those Space Invaders in all their orangey yellow glory). I don't remember the nature of any conversation about how I was doing at Uni, probably because my brain has deleted it on account of it not being that much fun. What I do recall is that I badgered my dad to have a go at Space Invaders on the VCS. He protested at first, claiming to be too old for such nonsense, but eventually I managed to press the VCS joystick into his hands.

I don't think I got it back for a few hours.

Despite the air of parental disappointment, I actually remember having quite a reasonable time that summer. My brother would come by from time to time, and we'd have games of Superman and Space Invaders on the VCS. We got super-expert at Superman, achieving sub-minute times in rounding all the criminals up and rebuilding the bridge whilst avoiding Lois Lane, kryptonite and that bloody helicopter, and when we weren't hunched over the VCS then as often as not we'd be down the Rec Soc playing Galaxian, Astro Fighter, pool, and "wocka wocka" – my brother's name for the new game with the little yellow ball that ate dots and was chased by ghosts.

I can't remember how it happened but I acquired a student - a lad of about 14 who was interested in learning to program – and occasional lessons were conducted in front of my ZX80. However, one day we happened to be out when the kid came around for his lesson, and bugger me if the little sod didn't break into the house and nick my ZX80 RAM pack and my VCS and all the games! I thought that was a

bit bloody harsh, but it made me resolve that whatever I ended up doing, I wasn't going to rip off anyone else to do it.

So I suppose I should thank the little bastard for that object lesson, but nonetheless it irked me greatly at the time, and I think I would have cheerfully pushed the little bastard's teeth down his throat for him if I'd seen him around after that, but funnily enough, he avoided me.

Over the course of the summer, I managed to trade my ZX80 for a nice new ZX81, and now that I didn't have to jump through hoops to get a stable moving display, I was able to produce a few simple games, including a version of our game Deflex that we'd made for the Commodore PET back in sixth form. One day I went up to London to go to one of the new 'ZX Microfairs' - gatherings of enthusiasts and vendors involved in what was proving to be a hobby of increasing popularity as more and more people got their hands on Sinclair's cheap little machines.

I wandered round, bought a copy of Quicksilver's Defender for the ZX81, and ogled the various bits of hardware and software available from the vendors. I was dribbling over the 16K RAM packs of one company, well out of my budget range at the time but nonetheless worthy of a bit of drooling, as I only had 4K and 16K seemed an astronomical amount of memory.

I got chatting to the bloke on the stand and happened to mention that I'd written a couple of games for my own 4K RAM ZX81. I fished out a tape with the games on and we loaded them there and then on a machine on his stand. To my utter amazement, he expressed an interest in actually buying some games off me to sell, and right there

he gave me one of the 16K RAM packs to work with! Nothing like that had ever happened to me before, and it was with great happiness and surprise that I returned home with a 16K RAM board that I couldn't have hoped to afford yet had got for free.

Thus equipped, my programming efforts redoubled, with the added satisfaction of thinking that now I was actually working - doing something I might make some money out of, or at the least use to score some more cool hardware. I started quite simply, polishing up Deflex and writing one of the inevitable 3D maze games (mine was called 3D3D because the maze was inside a cube, and you could go up and down as well as left, right, forward and back). But I wanted to do more than BASIC programs, so I acquired the sacred tomes of the hardcore ZX programmer - the ROM Disassembly and a good z80 machine code reference book. I came up with a version of Centipede, which was a coin-op game I hadn't actually played at that point, but which I figured I knew enough about to make an attempt at...

From *Sinclair User*, Issue 1, April 1982,

'Software Overview':



"The last arcade game at which I look at is DKtronics Centipede. I cannot describe it adequately without confusing all and sundry, so suffice to say it has excellent moving graphics and is very attractively presented, with the game becoming progressively more difficult in several ways.

"What is also very creditable about it is that the player can decide at the start how many bases each game should have and how fast the game should run. That means that several people can play at the same level of difficulty and the program produces a high score league table."

My first review!

Attending another ZX Microfair, I was standing in the queue outside and experienced my first moment of geeky fame. The two lads in front of me in the queue were talking about their favourite ZX81 games, and I heard them mention Centipede. There were a few versions around at the time, and I was curious to see if they had heard of mine. So I asked them "Have you seen Centipede by DKtronics?" They said that was the version they'd been talking about. I told them that actually I was the one who had written that game, and they said, "Wow, so you must be Jeff Minter!"

That was a very odd moment, but I was not dischuffed.

This minuscule foray into the emerging games biz showed no sign of actually being able to sustain me in terms of a career, however, so plans proceeded to get me back into further education so I could get a degree and a proper job in due course.

Eventually I found a place at Oxford Poly (Ox Poly as it was known to the denizens, and they had a rather excellent T-shirt with an ox logo that I regret not buying). It was, again, on a physics course but with an understanding that I would get to transfer to Comp Sci at the start of the second year if all went well. I attended Ox Poly while still living at my parents' house, which involved a rather hefty commute each day, by bicycle to the train station, then train to Reading, another train to Oxford, then another bike ride through town and up a steep hill to Ox Poly itself. Once there, I'd arrive in time to meet up with my fellow students for an hour, sitting round in the lobby and smoking occasional roll-ups, with perhaps a spliff between lectures and a pint of bitter and barley wine at lunchtime if the afternoon contained nothing

too stressful. There was a PET and a Research Machines 380Z there, and occasionally I'd bring in my ZX81 and show off some of the games I was making.

My days at that time began quite early, up before 7 to be on my way in time to arrive at Ox Poly by 9, and often not back until 8 or later in the evening. Once home, I'd retire to my bedroom and continue with my programming for the RAM pack company. The guy who ran the place came up with a hardware hack that, in conjunction with a tiny bit of code, allowed me to use alternate character sets on the ZX81. This would allow users to have character sets full of graphics characters like on the PET. This really appealed to me, as the ZX81's font was devoid of any game-oriented symbols at all, so I offered to design all the character graphics for the ROM hack - about 8 sets in all. If you examine the graphics in a DKtronics graphics ROM, you'll find the initials 'JCM' in one pair of characters. And another pair which is a little pair of llamas.

Preparing the ROM was quite a slow and painstaking business, involving drawing the characters onto graph paper and then converting the lines of dots into hexadecimal numbers which would then be programmed into the EEPROM burner that was used to make the chips.

One night I was sat in the living room with a graph paper pad on my knee doing this, and something very odd happened. I remember it quite clearly: my dad was watching Tenko on telly while I sat translating dot patterns into hex, and all of a sudden, for no reason at all, my heart started pounding at a great rate, as if I had just been

running as fast as I could. I felt quite peculiar and started sweating. Since I was sat at rest on the couch, this was pretty weird and a bit frightening, but after some minutes things returned to normal and I thought nothing more of it.

I carried on commuting to Ox Poly, working on the code at nights. As I got more involved - taking trips to Great Yarmouth on the train at weekends to work more closely with the guy who was making the graphics ROM - I bought myself one of the latest new things, a 'Game And Watch', where you had to control little LED firemen bouncing people out of a burning building and into an ambulance. It was made by some company I had never heard of called Nintendo. I played this a lot on the train during my commute to Ox Poly and journeys to Great Yarmouth. Soon, I began work on a version of Space Invaders using the graphics ROM, and it was shown to some acclaim at a ZX Microfair.

Something wasn't right, though. I began to feel uncharacteristically knackered and ill a lot of the time, and the instances of tachycardia became more frequent. I went to the doctor a couple of times, but he refused to believe that there was anything wrong with me and sent me home with instructions to have a hot bath. The general consensus was that it was all in my head, perhaps a byproduct of stress from all the work and commuting.

But my mum had found some entry in a medical dictionary pertaining to "disorderly behaviour of the heart", a psychological condition apparently suffered by soldiers at wartime, and brought on by stress. She showed me the entry one morning when I was getting up to go to Ox Poly, and, fed up of feeling ill all the time, I decided that if it

was all in my mind I'd prove it to myself by damn well exerting myself to the full on the journey to poly.

Accordingly I rode my bike as fast as I could to the train station and ignored the feelings of discomfort and nausea I felt upon arriving there. I took my trains, raced through town and pelted up the hill to poly, parked my bike and went and sat down with the guys in the lobby.

There's a very definite moment when you realise that something is seriously awry, and curiously enough it isn't accompanied by panic - just the certain knowledge that you have to start doing something now to help yourself. As I sat there, the world seemed to recede down a long tunnel, and all I could see was a spot in the middle. I felt great discomfort in my chest, and after a minute or so, although I could see properly again, I felt terribly, desperately weak. I apologised to everyone and left – using my bike as a crutch – and got myself to the train station, where I phoned my parents and asked them to come and collect me at the other end, because I was certainly not capable of making my way back from the station by myself.

My dad ended up carrying me and my bike back to the car, and I was taken to the doctors' where they finally hooked me up to the machine that goes ping, and discovered that actually, no, it hadn't all been in my head after all.

And that was the end of my further education career, and at the time it was horrible and terrifying. But that unfortunate incident left the way clear for the start of something else.

Part 7: A Beginning

The history of Llamasoft doesn't only belong to me; it belongs to many people who were caught up in the whole thing - the people who bought the software, the people involved in producing the games - and also friends and family who helped out and without whom there would be no history to tell. It belongs also to my parents, whose lives were occasionally turned upside down by all the goings-on, and who can have had no inkling of what was about to happen when I started staying late nights doing odd things at sixth form college.

My mum, in particular, was instrumental in protecting me from some rather less-than-scrupulous characters I encountered in those early days, eventually throwing her lot in with me and becoming a

partner in Llamasoft; and my dad, despite not being into videogames at all before I tainted him with the 2600, became fascinated with the whole business of both creating and playing videogames, and co-designed games with me and was a gamer for the rest of his life.

My dad was also a good diarist, and spent a good deal of time over the years working on writing a family history, informed by notes in his diaries going back several decades. This he eventually printed and distributed to me and all my brothers - it was an excellent and interesting read, particularly interesting to read his own perspective on events that we all only vaguely remembered from childhood or as distant family lore, from a time before we were born.

He also kept notes over the years about the happenings during the Llamasoft era, and he'd been planning to write a History of Llamasoft himself. Unfortunately he never lived long enough to do that - but upon going through his old stuff on my mum's Macintosh after his death, I was moved to discover all of the notes that he'd made for the project. Seeing his notes made me realise that there really was an interesting tale to tell about those old days, and that by doing the History myself, I could complete what he started, and in doing so actually engage in one last collaboration with him.

The notes are by no means a complete History - some have been more or less completely written up, others are just stubs and ideas and bits where he was going to ask me to fill in some details of my own later. As we are now getting to the interesting part of the History, this seems to be the right time to introduce his notes and commentaries. Wherever possible I'll quote his actual writing, and I'll expand on his notes with my own thoughts and memories as we go along.

I'll begin with his introductory paragraphs, which cover the same ground as I already have in preceding chapters, to recap and bring us up to date.

Pat Minter:

A History of Llamasoft - Pre-History & Beginning

Llamasoft has added a lot to our lives since it was first established in 1982, so I feel that it is worth setting down an account of its triumphs and adventures.

Of course Jeff was the creator and instigator of the whole thing; if he hadn't been so determined and so confident in his own abilities, Llamasoft would not have happened and Hazel and I would have a much calmer and more vegetative decade or so.

Jeff showed no apparent interest in computers until, at sixteen, soon after starting his A level course at Queen Mary's Sixth Form College in Basingstoke, he discovered that there was a Commodore Pet on the premises. The Pet was the first practical and seriously marketed "personal" microcomputer and simple games were beginning to appear for it, mostly created by students - I guess as a diversion from the mundane business of actually studying.

As this interest had not flowered until after his A levels courses had begun, Jeff had to progress by self-tuition, driven by his newly acquired enthusiasm. The first sign, for us, that something significant was afoot was when he started to leave home early to go to college- a striking departure from the norm. In fact he was ensuring that he could spend time on the computer system before the "authorised" students arrived...

In 1980, having achieved satisfactory A level grades in Maths and Science, he was accepted for a place at the University of East Anglia in Norwich and in the September was installed there to begin a 'Physics and Computing' degree. This course was intended to comprise 80% physics and 20% computing. Jeff approximately reversed these percentages and in doing so had a fine and instructive time, to the extent that after a year he was asked to leave.

We were rather worried by this last development and after discussing matters with him we began to search for a course which would lead to a degree in Computing, which we felt would better match his interests and improve his motivation. Various courses were considered- I remember an interview at Portsmouth Polytech, for instance, but eventually he was offered what seemed a suitable place at Oxford Polytechnic.

A Crisis

Having obtained a place, the question of accommodation had to be resolved. This proved to be difficult so it was agreed that initially he would commute the 35 miles to Oxford each working day. He would cycle to the station at Mortimer (4 miles?), proceed by train to Oxford, which entailed changing trains in Reading, cycle to the college and return by the same route. This was clearly a physically demanding schedule but then he was young and fit and he had a decent bike, didn't he?

The course began in early October and seemed to suit him. He travelled to and fro without undue complaint and seemed to have settled well into the travel routine. Sometime during this period Jeff came into contact with the head of a company that was trying to establish itself in the rapidly expanding home computer market. This

gentleman was interested in the games Jeff had already developed on his Sinclair ZX81 computer (purchased with the proceeds of a part-time office cleaning job). He invited Jeff to his headquarters in Great Yarmouth and a couple of projects were agreed which Jeff would work on in his spare time.

So Jeff now had a place on a congenial degree course and the possibility of earning some extra cash while improving his practical knowledge of computer programming. At the beginning of November, on a Saturday afternoon when the house was empty, there was a break-in and Jeff's ZX81 computer and a few related items, were stolen. The culprit turned out to be a kid who Jeff had been coaching in computer technology! Eventually the stolen goods were recovered and returned by the police.

On November 23rd we had a phone call from Jeff, from Oxford- he said that something was wrong- he was unable to pedal his bike. Was the bicycle defective in some way? No, he had pains in his chest. Greatly alarmed we asked if he could make it to Mortimer; he thought that he could and I drove to Mortimer station to meet him. Obviously the next step was to consult our local doctor. He in turn called in a heart specialist, Mr Fowler, who came to our house that evening. ECGs were made and a diagnosis was soon forthcoming- Jeff had Pericarditis, an inflammation of the lining of the heart, a condition sometimes experienced by athletes and which if not recognised and treated promptly, could be fatal.

Of course it didn't turn out to be fatal, or I wouldn't be sitting here writing the History at all. But it was extremely scary for everybody involved. I remember waking up one morning and finding my brother holding my wrist looking for a pulse, because I'd slept so late that my mum was scared that I'd died in the night.

As is the nature of such things, it was a lot more scary when I didn't know what was happening than it was once the correct diagnosis was made. Which isn't to say that it wasn't terrifying at times - some nights I'd get a panic on and be sure I was going to snuff it imminently, which I would usually assuage by doing some programming. Programming was an excellent way to take your mind off scary thoughts and worries.

There wasn't any real form of treatment for the condition other than complete rest - the heart became 'distressed' by rubbing against the inflamed pericardium, and all one could really do was chill out as much as possible until hopefully the inflammation went away and things got back to normal. I wasn't so desperately ill as to need hospitalisation, and it was decided that I would be less stressed to be at home. The condition did leave me feeling very weak a lot of the time, and for some weeks I was unable to do much that couldn't be done in a lying or sitting position.

One of my last acts before collapsing in Oxford had been to badger the Commodore dealer there about availability of the new Commodore machine that was just being released, the Commodore VIC-20. I was particularly keen to get this new machine, since not only did it have a much more robust keyboard and case than the rather flimsy ZX-81, it was also the much more game-oriented offspring of the old Commodore PET series that I'd cut my coding teeth on. It had sound and colour (and not a great deal of memory, but still more than a stock ZX-81) and I anticipated being able to write games on it that at last came close to capturing something of the arcade experience.

A couple of weeks after the incident and my diagnosis, my mum took me into Basingstoke to go shopping. Not that I could actually do any shopping - I just sat in the car, too weak to even walk the streets of Basingstoke. But I was able to send her up to the computer shop at the top of town to ascertain the availability of Vic-20s, and she returned to the car park with a Vic-20 package, which I lovingly cradled on my knees all the way home, looking at the pictures of happy families and chunky pixel sailing boats that adorned the outside of the package.

The VIC was a lot like the PET I'd started on, as I discovered when I got the machine home and hooked it up to the family telly in the living room, where I could lie comfortably and fiddle with it. The BASIC language was the same; the screen had fewer characters per line, but that was more than made up for by the fact that each character could have one of eight colours. In addition, one wasn't limited to the PET character set - one could sacrifice a little bit of RAM and use it to generate any character set you liked, just as in the graphics ROM project I'd been involved with, but with the advantage that one didn't need additional hardware, and that being in RAM, one wasn't limited to a fixed number of alternate sets. Anything you could think of you could define.

The VIC also had a splendidly noisy sound generator - no more arcane tricks getting sounds out of AM radios for me! - and a marvelously chunky and robust keyboard which was a real pleasure to use after the fiddliness of the PET's tiny calculator-style keyboard and the unresponsive membrane efforts on the ZX-80 and ZX-81.

I spent the most part of that strange and stressful December lying in front of the VIC and the family telly, learning my way around. I learned where the screen RAM was and how to do the character-defining thing - in fact I wrote my own little program to let me edit the characters more conveniently. I wrote some simple games - a conversion of the 'Deflex' game that we'd originally made on the PET, except with noisy crashing sound effects and flashing colours when you hit a target, a Breakout game, a version of the hoary old City Bomber game that everybody used to do. As my dad recalls it:

Jeff had now acquired a Commodore VIC 20 computer as well. This had the advantages of a "proper" keyboard, colour output and a 3.5k of memory, a spectacular advance over the ZX81's 1k!

His first game for the VIC 20, which he soon produced, was ROX, programmed in the Commodore version of the BASIC computer language. Rox presented a simulation of a planetary surface on which stood a space station which the player had to defend by firing missiles at descending meteors. It was very simple, colourful and it was fun to play.

Christmas was rather a subdued affair that year though I recall that Jeff was able to play Rox, lying prone, using the living room telly as a display.

Rox, for all it was a very simple game, was something a bit special to me. I'd made those other few games on the Vic, and they were fine, but were all simple and all things that had been made before. Rox was something else, a little game that I made up myself, redefining the VIC characters to create a lunar surface upon which a little lander

descended, and which was then beset by falling meteors. There were only a few controls - to fire off intercepting rockets in three directions, or to panic and leave the surface, ending the game. Successive meteor impacts chipped away at the lunar surface, and if the surface was breached, or a meteor hit the lander, then you'd lose.

It was a really simple game, really all down simply to choosing which rocket to fire and timing it right - but there was something about it. Me and my dad played it a lot that Christmas, and I refined it, adding successive waves of meteors and occasionally an enemy missile strike to contend with too. I made it look nicer than any game I'd made before, designing a custom character set so that even the text appeared in a futuristic style, and adding a proper hi-score table and everything.

Nothing clever, only BASIC code, but it looked good for a game of its era, and... it played well.

We kept coming back for more.

Anyway, let's resume my dad's narrative...

Recovery

Happily the New Year (1982) saw a slow but steady improvement in Jeff's health, albeit with one or two setbacks. The medics kept a keen eye on him, with regular ECG examinations to check the current state of his heart function.

The chap from Great Yarmouth was anxious to capitalise on Jeff's talents and Jeff was keen to respond.

Consequently, in spite of our misgivings he went to Great Yarmouth at the end of January to complete some work for the fellow's company. This resulted in the marketing by that company of Jeff's version of the established arcade game Centipede, and also a program which would give the ZX81 computer higher-resolution graphics. The company must subsequently have made a great deal of money from these two products- they were advertised widely in the new computer games magazines- but Jeff's reward was derisory and it was not proposed to pay him any royalties at all for the hi-res program, which was unique and commercially valuable. Its value is illustrated by the fact that the bloke's cousin, who worked for him, defected to start his own company, selling thinly-disguised versions of the same products, including Jeff's hi-res program and of course no money at all was forthcoming from that source (although I did tackle the scumbag at a computer show some months later - he offered a payment of 10 pounds to which I made a rude reply). We subsequently arranged for him to receive a solicitor's letter on the subject but were keenly aware that our financial resources were meagre and that at that time software piracy was not legally recognised.

Six months later, after many letters, phone calls and unfulfilled promises, the guy paid a once-for-all sum of 500 pounds for the hi-res program, which must have been a fraction of a percentage point of his profit from it-booming times in the computer games business.

Jeff returned from Great Yarmouth in February in somewhat worse shape than when he had left home a week or so before. Fortunately the setback was shortlived and his health continued slowly to improve.

Here my dad is remembering a combination of events, some of which happened before I became ill - starting to work on the

"Graphics Rom" project and Centipede for the ZX-81 - and what happened afterwards.

The guy from Great Yarmouth I'd been working with was an ex-used-car salesman, a great corpulent slug of a man whose business practices turned out to be pretty shady, at least back at that time. Remember, at that point I still had no idea that I could ever actually make my own way selling and making games, and the small amount of biz I'd done with the man led me to believe that, as I was seemingly decisively out of the educational system due to a combination of flunkage at Norwich and illness at Ox Poly, pursuing work in league with him was probably the best bet for me to get off the dole and into employment.

Therefore, once I was recovered enough (barely) to travel, I set out with the intention of actually working for the man in Great Yarmouth. I got on a train and headed up there, booking into a cheap B&B when I got there. I still wasn't feeling at all great, but I figured I could hack sitting at a desk coding and flopping out in the B&B at night. The first day I was there he sat me down in front of what I think was an MZ-80K or derivative, a Japanese machine somewhat like the old Commodore PET, with the intention of having me write business software for his company. I remember sitting in the guy's office, in some warehouse that he'd recently acquired when his own biz got too big to run from his upstairs bedroom, playing with the Sharp and actually thinking about how to write business software in Sharp BASIC. At nights I'd get some chips from the local chippy (which had a Space Invaders machine in, at which I'd occasionally essay a wan game) then go back to my room at the B&B and pore over "Vic

Revealed", a red, paperback tome which contained in-depth secrets and deep knowledge about the workings of the VIC-20.

I wasn't there for long. One day I decided that I really needed to ask the man about what my rewards were going to be for my part in the graphics ROM project - it didn't seem unrealistic to expect something, given that despite my encroaching illness I'd worked pretty damn hard putting things together and writing software for the thing.

The guy basically turned around and told me that I'd be getting nothing at all for it. As far as he was concerned it was now a hardware thing for him to sell, and that was an end of it, despite the fact that all of the actual contents of the ROM had come from me. He thought I was doing it all "as a favour".

That was the end of any plans I had to work there, to be certain. I became very upset, and at that time such upset caused me a good degree of physiological distress too, thanks to my illness. I arrived gasping and shaking back at the B&B, and the kindly lady who ran the place found me white and in tears, packing up my stuff. I felt so frustrated, and angry at myself - angry that I had been taken in and used so, angry at my stupid illness that was making me feel like shit for months on end, just completely upset, and ill. I made my apologies to the lady and headed for the train station to get myself home where I arrived physically and emotionally shattered.

It's true about the bloke's cousin, too - shortly after I'd started doing ZX81 programs for the man, another company with almost the same name started up, and they were selling pirated versions of my own programs - identical code except for my name crudely hacked off the title page and replaced with something else. They were both

repeatedly challenged about this but there was never a straight answer, they claimed that there was no relation between the companies and what they sold, despite them being cousins. It warmed my cockles and made me proud when my dad told him to piss off with his paltry settlement offer at that show some months later. We did extract a desultory amount from them in the end but by then we figured it best to be happy we even got that out of them, and just move on.

And that was the end of the lot from Great Yarmouth.

(Many, many years later, trawling through unfamiliar program images on a Vic-20 emulator, I came across something surprising. I was loading a .tap image of some US game, and... something seemed familiar, something I hadn't seen for 20 years... the way the character images loaded from the tape, gradually filling the title screen with a familiar font... little llamas on the instruction pages... but the title screen had plainly claimed that the game originated in some US state...

...then I remembered: before the irreconcilable breakdown with the Yarmouth guy, I'd written him a handful of Vic-20 games, really simple ones, which he'd wanted to release as a cheap pack of games to sell for the VIC, opening up that market alongside his existing ZX81 biz. Ten games on a tape or something like that, and one of them was "Space Zap", based on a description of an obscure arcade game I'd never even seen.

And it was "Space Zap", llamas and all, that I saw on that obscure US tape image. It'd plainly been taken and sold in the US, completely unknown to me, only finding out about it 20 years later.

Probably the US guy just picked it up and copied it himself, from the tape sold legitimately by the Great Yarmouth guy, and hacked the title screen and sold it himself, and I'm just being paranoid to think that slug-man did an export deal on the sly.

Probably...)

Back to my dad's narrative (I hasten to add that my dad would never have used the term "Smeghead". However in his text he used actual names, and I have grudgingly changed those to protect the guilty throughout):

The Smeghead Period

Through February and March, to judge from the information to hand, Jeff lay low. In fact I feel sure that he was working- and further fruits of his programming were shortly to appear. Also sometime during this period, probably at one of the computer shows now being held with increasing frequency, mainly in London, he made the acquaintance of Smeghead. Coincidentally Smeghead also lived in Tadley. He was seventeen and lived with his parents. His father was a staff sergeant (or some such) in the Army and owned a small furniture removal business.

Over the next weeks Smeghead became a frequent visitor. Although he and Jeff shared an interest in computer games, Smeghead clearly saw himself as a budding business tycoon and computer games as the medium for achieving this ambition, whereas Jeff's interests were in conception, design and programming. Their possible combination in some sort of partnership seemed logical and potentially profitable. Although Smeghead was young

he was backed by his father who obviously saw this possibility as a means of extending his own business interests, particularly as he was shortly to resign/retire from the army.

My diary for 21st March 1982 contains the one word 'LLAMAS!' from which I deduce that this was the day that Llamasoft was formed.

Actually I think I met Smeghead through a mutual acquaintance, a guy who lived on the same estate who also had a computer (I think it was a ZX81 back then, although he later got an 8-bit Atari). Smeghead lived round the corner from this guy; the family ran a video rentals business from out of their garage. Anyway, of an evening I frequently used to wander down to the estate and hang out with Smeghead and my mate, or they'd come back to mine, and Smeghead started getting the idea that instead of flogging my stuff through the likes of the Great Yarmouth bloke, maybe I should think of doing it myself.

I must admit that similar thoughts had been occurring to me. The "games biz" had sprung into startled existence in the space of a few months, and the poor thing was in sorry shape indeed, not really knowing what to do with itself. People suddenly realised there was a market selling programs for the newly-fashionable "home micros" and the result was an absolute shambles. Whether peddled by individuals working from their bedrooms or hastily-formed companies formed to be the first to exploit this new scene, the software on sale was almost uniformly execrable, usually artless apings of popular arcade games cobbled together in BASIC or, if you were really lucky, "100% machine code!".

It was after having paid seven quid for a version of "Asteroids" so mind-numbingly, gut-wrenchingly and unremittingly awful as to beggar belief that I started thinking to myself "bloody hell, even I could do better than that" and wondering if that was indeed what I should do.

In the meanwhile I carried on exploring my VIC-20. I got myself an 8K RAM expansion - which was sheer luxury, and allowed me to try more ambitious things on the little machine. I also bought myself a large, metal, Welsh expansion unit, and a "Spot Assembler" cartridge, which was a tool that let me examine memory in the form of an assembly language listing, and enter opcodes as mnemonics rather than do all that tedious mucking around in hexadecimal.

Something I was playing around with was the concept of making a big, virtual screen inside the VIC, and then being able to scroll over the virtual screen at will - the default VIC screen display was only 22 characters wide, which was a bit restrictive. I wanted something that would let me work in a larger area.

I wrote some assembler routines to handle the display and scrolling of the virtual screens, and it all worked pretty well in the end, but I wasn't quite sure what I should do with it. It was a neat hack in need of something to be used for.

I also worked a bit on the little tool I'd made to help me design graphics and character sets for the VIC. I extended it so that you could put a cluster of characters together and use those as a tiny hi-res display for making larger designs - you could push a little cursor round the display area and "draw" inside it using the keyboard.

One day I was home alone one afternoon, lying on the floor idly playing with the graphic design tool, and thinking about animals. I pushed the cursor round and drew the outline of a llama. I sat and swigged my tea, looking at the llama, and then, on a whim, directed the cursor to some empty space below the image of the beast, and wrote there, in tiny letters, the word "Llamasoft!!!" (Goat help me, multiple exclamation marks and all. Forgive me. I was young).



The ur-Llamasoft logo, complete with THREE!!! exclamation marks.

Satisfied with my work, and needing to nip out to get some money, I finished my tea, cleared the screen, and then used the keyboard to reconstruct my little llama logo in the middle of the screen. Then I typed beneath it "GONE TO BANK, BACK SOON" and left the telly on as I went out, to explain my absence should my mum return while I was away.

It was more an idle doodle at that time rather than a proper intent - but Smeghead thought there was mileage in it, and encouraged me in such thoughts at every opportunity - with the proviso that he would be my representative and promotor, and handle all the business side of things, whilst I would be the creative engine actually producing the goods. He got me round to his dad's house on a couple of occasions to talk about it.

They were recent arrivals in the town, having been over in Northern Ireland owing to Smeghead's dad being in the army. They seemed to be doing OK, with irons in a few fires between Smeghead and his dad, and were almost well-off in a somewhat chavvy, slightly shady kind of way. His dad was an odd character, a short, strident man who reminded me of a strutting Bantam cock. It did make me squirm a few times when I was up there and I heard them talking disparagingly about "Fenians" and "pakis" - I'd never been around anyone with views like that before. I guess I put up with it because I figured hell, maybe being in the army and in a place where you might get shot or blown up makes you less tolerant, or something like that. But it still made me uncomfortable.

Between them they did their best to encourage me to allow them to run the business side of things if I did decide to start selling my own games. They made great play of the fact that I was still recovering from my illness, telling me that they would do all the hard work and shield me from the need to worry about the business side of things. All I had to do was rest, recuperate, and write code, and they would reward me with a whole 30% of anything that they made selling it.

Goat help me, I could cheerfully travel back in time and knee my 19-year-old self firmly in the bollocks for being so easily led up the

garden path, round the houses and into the barn with all the cows in, but back then I was young and stupid and hadn't the first idea how to go about any kind of business activities at all, so I fell for it.

We decided that we would start by placing a small advert, printed out on the Vic-20 printer, for some of the early games I'd made already, and see what happened.

And we decided that we should call ourselves "Llamasoft".

Thankfully, Smeghead and his strident dad had figured without allowing for the level head and fiercely protective nature of my mum...

Part 8: The Dawn of Llamasoft

If we were to call ourselves Llamasoft, we actually needed some software to sell, and so during the summer of 1982 I busied myself with the creation of what were to be the first Llamasoft games for the Vic-20. I'd already got a couple of simple games that I'd done with a view to letting the Great Yarmouth chap sell, and so those were part of the initial lineup. One of them was a simple 3D maze game of the kind that everyone used to do back then, with redefined characters which put bricks on the maze walls along with occasional Pink Floyd hammers and the Llamasoft logo that I'd created with my character editor. Another was – well, probably one of the most shameful things in Llamasoft history, a joke which got out of hand...

One of the first games I'd written on the Vic was a reworking of one which we'd had on the PET in school, and indeed every machine of that era must have had a version available as a type-in listing; it was one of those canonical simple BASIC games that everybody and their dog did back then. It was a version of the old "City Bomber" game – the computer draws a "city" at the bottom of the screen and a plane that slowly crosses the screen at the top, descending one line each time it reaches a screen edge. The sole player control is a "Bomb" button which, when pressed, causes a single bomb to drop from the plane, fall down and destroy a chunk of the buildings below. The player needs to carefully time the dropping of the bombs in order to raze the entire city, allowing the plane to land on the ground, whereupon the next level begins with slightly higher buildings.

Now, at the time, unfortunately there was conflict between Argentina and Britain over the sovereignty of the Falkland Islands and... just for a joke, you see... I... created a little graphic of a waving Argentinian flag and stuck that on top of the buildings in my city. And made the Vic play "Rule Britannia" when the plane landed successfully. And called the game "Bomb Buenos Aires"...

I swear, it was all done in what was intended as a tongue in cheek manner; in real life I was not particularly fond of the mess that Maggie seemed to be getting us into, and I certainly didn't believe that it was worth anybody actually getting killed over, on either side. But I guess I thought that it might get us noticed, in a tasteless kind of a way, and indeed it did – as my dad tells it:

Jeff produced his next VIC 20 game, BOMB BUENOS AIRES!. The title shamelessly capitalised on the fact that

the country was in the midst of the Falklands War with Argentina. Distribution was by mail order after copies had been sent to magazines and to the press to generate publicity, and of course by sales at computer shows.

Hence, from the Daily Telegraph of 5th June 1982...

“Do you fancy being at the controls of a Vulcan bomber and releasing thousands of tons of bombs on Buenos Aires? This experience, with accompanying thunderous explosions and the playing of Rule Britannia may be had on a jolly new computer programme unveiled in the Commodore computer show at the Cunard Hotel, Hammersmith, by author Jeff Minter and publishers, Lamma Software of Tadley, Basingstoke.”

The controversial title and subject clearly had had an effect- complaints were made to, and an official admonishment was eventually received from, the Press Complaints Commission and some months later Llamasoft received its first accolade, the 1982 Bad Taste award!

We decided to respond to the complaints by re-issuing the game under the less contentious title, BOMBER and a letter of apology was sent to the Complaints Commission. Another version of this game, entitled BLITZKRIEG, was produced for the Sinclair Spectrum computer (successor to the ZX81).

To be honest having the Torygraph ranting on about releasing thousands of tons of bombs pushed the whole thing too far into the domain of tastelessness for me, and I was quick to remove the silly references to the Argentine conflict and revert the game to decent anonymity.

We had a couple of other simple Vic-20 games to sell too - “Rox”, which I have already mentioned, and a game called “Headbanger's Heaven”. This was a reworking of another popular game of the time, usually called “Moneybags”, in which you controlled a little guy at the bottom of the screen who had to walk to right hand edge of the screen, retrieve a bag of money (usually represented by a simple ASCII “\$” character) and stash it at the left side of the screen. His journey was endangered by missiles that fell in a continuous rain from the sky. Shelters akin to the houses in Space Invaders afforded some protection from the rain of missiles, but these were continuously eroded, and eventually the guy would sustain a hit and lose a life. Simple stuff.

In “Headbanger's Heaven” I modified the way it worked a bit. The missiles were replaced by a rain of hammers (cue another excuse to use my Pink Floyd hammers as a logo) and the little guy was named Chico, after a heavy-metal-fan friend of mine from University (the same one who had achieved such awesome mastery of Space Invaders). Chico still had to collect the money bags, but the twist was that, as a heavy metal fan, he actually enjoyed being hit on the head by the hammers (“headbanging”, you see). However in order to score properly the hammers had to hit him directly on the head – a blow to the body caused a loss of life as normal.

Correctly nuttered hammers would cause Chico's “pain meter” to rise, and the higher the meter, the more points he received for moneybags collected. If he sustained too much pain, however, he'd lose a life; the trick to scoring big points was to max out his pain meter and then look for an aspirin (the rain of hammers also helpfully contained the occasional aspirin pill). An aspirin to the head with a

maxed-out pain meter granted the player a useful bonus, reset the pain meter and also rebuilt the protective barriers, allowing him to play on and build up the pain meter again for even more bonus points.

We placed a few cheap adverts in the back of computer magazines of the day (and we're not talking full-page ads at this point, it was just a small square in the reader ads section, printed out on a Vic-20 dot matrix printer) and in due course began to sell some of these simple games. Such games were pretty much small beer though – simple BASIC hacks such as anyone might do, with little to distinguish them (apart from in a bad way in the case of Bomb Buenos Aires). They sold mainly because there were very few people selling games at all at that point – certainly there wasn't much in them to truly distinguish Llamasoft as being significantly different or better than all the other people doing exactly the same kind of stuff.

We needed some kind of “magnum opus”, something that would hopefully let us claim some kind of significance in the emerging biz. And back then, in that dawn of time, the way to do that was to produce a version of a popular arcade game, and to do it in machine code.

A year or so previously, before I'd become ill, I experienced something of a videogaming epiphany on a trip to Southampton to visit my gran. I'd gone to a travelling fun fair that was set up that day on Southampton common and, as was my habit under such circumstances, I'd gravitated towards the arcade tent, thinking to enjoy a few rounds of Space Invaders or Galaxians. However I never got further than the tent

entrance. There, out front and to the right of the entrance, stood a new arcade game which captivated me from the first glance at the graphics and the first hearing of the sound effects. This monster looked and sounded like no other. The controls consisted of a joystick and a bewildering array of buttons. The graphics were sharp and colourful, and when your ship moved the whole world moved with it too. The world was actually bigger than the screen, and there was even a tiny radar enabling you to track events that might be happening off-screen away from where you were that instant. You controlled a tiny fighter plane, and your mission was to rescue “Humanoids” from dastardly enemies who would kidnap them and mutate them into sick, shaking Mutant craft that would spew deadly accurate shots and think nothing of ramming into your ship at high velocity.

That game was Williams' “Defender”, the first coin-op game from a programmer whose name I didn't know at the time but who I would come to respect and revere as one of the best videogame designers ever – Eugene Jarvis. Defender was awesome in so many ways. The scrolling game world in which you could fly wherever you wanted to was a first in videogame design, as was the radar. Whereas most games depicted explosions as a simplistic, static sprite, things that blew up in Defender really blew up, into hundreds of brightly-coloured shreds which streaked across the screen. When the game got busy, it was like flying through the night sky on the 5th of November. The sound effects were deep, menacing, intimidating... and the gameplay was iron-hard. My first few goes in front of that bewildering button array saw my arse handed to me in short order, again and again – but I couldn't tear myself away, not until I had achieved at least a very basic level of competence, and also run out of money.

I remember walking back to my gran's place thinking about Defender and about how, one day, I would love to make a game even a tenth as awesome as that.

And so, that first summer of Llamasoft, Defender sprang naturally to mind when I was considering what I should code as Llamasoft's first, seminal, “proper” game.

Now, Defender was a tall order for the machines of the day, and especially for a Vic-20 in my inexperienced hands. For one thing, with that large game world I was going to forget doing it on the unexpanded Vic. My simplistic, brute-force approach to maintenance and display of the multiple-screen game world would require me to make the game for a Vic with 8K memory expansion.

I already had some scroll routines that I'd written in my experiments with “virtual screens” larger than the physical display of the Vic, and these were pressed into service to implement Defender's scrolling planet. They weren't elegant, fine scroll routines like the real Defender; they scrolled in chunks of 8 pixels at the time, the width of one Vic-20 character, and at low speeds the motion was decidedly clunky. The spaceship I made was horrid, at least six times too large, and no matter how much I fuffed about I couldn't smoothly replicate the fluid way that a Defender ship turned and manoeuvred. My big clunky ship handled like a bus on a skidpan by comparison. My lasers didn't fade away pixel-by-pixel as real Defender lasers – they were big ugly lines that made a hideous noise when you fired them, like a robot gargling nuts and bolts. My Humanoids would occasionally glitch and

leave chunks of mountain stuck in mid-air when you managed to rescue one that was falling.

In other words, “Defenda”, as I called it, was a bit shit.

However, back then, *everything* was a bit shit, and there was the possibility that Defenda might be marginally less shit than some of the other games out there. And games in machine code for the Vic were rare.

So that summer I toiled to finish my horrid version of Defender, in order to have the game finished before the nascent Llamasoft went to its first ever computer show as an exhibitor – a Commodore show at the Cunard Hotel in Hammersmith. We'd booked just about the cheapest space possible, really just room for a trestle table, a telly and a Vic to show the games on. I finished Defenda with a few days to spare, and those days were spent, I remember, manually duplicating copies of the cassette tape with the game on, my mum and family helping with the business of folding up the inserts we'd had printed at a local printers', applying sticky “Llamasoft” labels to the cassettes, and packing them away into plastic cassette boxes. I think we did a couple of hundred of them before it was time to go to the show, and we had no idea if we'd be bringing them all home again.

I was still recovering from my illness and remember feeling pretty unwell at the start of that show. We went up early in the morning with Smeghead, and got our trestle table set up, just around the corner from the main Commodore stand. During the show itself the stand was manned by me and Smeghead alone – there really wasn't room for anyone else. We had a telly and a single Vic into which

“Defenda” was loaded, and a joystick placed so that prospective customers could sample the bus-on-a-skidpan, robot-bolt-gargling gameplay for themselves. I remember wondering what the hell would happen when the doors opened and actual gamers came in.

I remember it was busy. The home computer boom was just starting to hit its stride, and there were a lot of Commodore enthusiasts there that day, and on all the three days of the show. Much to my amazement (and delight), we ran out of tapes on the first day. There were frantic phone calls back home, where the duplication chain was cranked up again for an emergency run to produce more tapes for the remaining days. It seemed that although Defenda was palpably cack, it was actually marginally less cack than some of the other games there, and so we sold tapes pretty much hand over fist the whole time. This was rather good.

Another, entirely unforeseen, rather good thing happened at that show. A smartly-dressed American gentleman in a suit approached me on the stand one day and started asking me questions about Defenda. It turned out that he was from a US software house who were themselves getting into the games market and were at the show looking for potential titles to distribute in the US, and his eye had been taken by my clunky Defenda. He started asking me if I was interested in doing a deal (yes, yes I was!) and if I could put the game on a ROM cartridge that would work in the unexpanded Vic (I had no idea how the hell to go about producing a ROM, but I said that yes, I could, and trusted I'd find out how later).

On the last day, after the show was over, we brought this charming gentleman, whose name was Jay Balakrishnan, home to meet my Mum, who was, I believe, quite surprised. He opened his smart briefcase and began to tell us about a contract to produce my Defenda game for his company in the US, an outfit called Human Engineered Software, or HES for short. Provided I could produce the game on ROM, and provided that he was true to his word (and he was), Llamasoft was set to earn a not insignificant amount of money from this deal, certainly far more than our domestic market would provide.

That first Commodore show then was a remarkable success for Llamasoft in many ways. But with regard to the relationship between me and Smeghead – the wheels were just about to fall off.

Trouble had been brewing there for a while anyway. The primary factor was that Smeghead and his dad wanted to take 70% of the profits from Llamasoft (and I don't know quite how they worked that out, since it was me writing all the software) and I was uncomfortable with that and my mum was definitely having none of it, insisting that if we were to become a partnership it would have to be 50:50, them and us, or not at all. There were other factors too – Smeghead wanted to find software from other people too, and sell that under the Llamasoft name; but when I looked at the kind of stuff he wanted to sell, it was obvious that he didn't really care if what we sold was absolute rubbish a hundred times worse than even my clunky Defenda, and I definitely didn't want to do that. Just because you could get away with selling any old rubbish back then doesn't mean that you should. So I put my hoof down and insisted that Llamasoft would sell only my games, and they didn't like that much at all.

And so, throughout the rest of that summer, things continued, with the partnership between me and the Smegheads getting more and more uncomfortable. However we had a couple more computer shows lined up to go to, and more software was needed, so I tried to put all that aside and get on with what I did best – writing more code.

I obtained a Sinclair Spectrum to go alongside my Vic-20, and although the Speccy was never really to become the platform of choice for Llamasoft, I set about producing a few simple games for it just to allow us to dip a toe in that market. I did a port of that silly City Bomber game (no question of any Argentinian flags in the Speccy version, I wasn't going near that particular stuff again) and implemented Headbanger's Heaven on there too (curiously enough these days Headbanger's Heaven is easily found for Spectrum emulators but the superior Vic version is nowhere to be found). I also did an updated version of my old PET favourite Deflex, which in the intervening years I'd also programmed versions of for the ZX81 and Vic. The Speccy version was probably the nicest of all of them, with more complex layouts for the various levels, and a large animated llama that would walk across the screen to clear a level if the player took too long to finish it himself.

I also set about creating a couple more 8K games for the Vic. One of these was a fairly crude version of the “Painter” style games that were popular back then, in which the player manipulates a character that moves around a grid, avoiding enemies and attempting to “paint” all the edges of the squares of the grid. This was inspired by a

few nights' drinking at the Fighting Cocks pub in Tadley, where they had an Amidar tabletop machine at which I acquired a reasonable degree of competence (and which I was somewhat peeved about when it was removed and replaced with some weird thing with this odd little guy running about on girders and a big monkey). Consequently my painter game, called "Traxx", featured my first ever interrupt-driven music routine, which continuously played the tune from Amidar while you played.

Traxx again was rather clunky – maze games are less than ideal to control with an 8-way joystick, but a lot of the awkwardness of controlling it was down to my own inexperience too, I still hadn't found the knack of making things handle well in a game. Still, it was again probably marginally less shit than some of the stuff out there, and so it didn't do too badly.

The other Vic game was one of my first attempts at something approaching an original game, and to be brutally honest it ended up being a bit of a dog's breakfast. You controlled a little man with a hammer, and walked upon a surface below which were little guys with pointy sticks. These little guys would occasionally poke their pointy sticks up through the surface and if you happened to be standing above them you'd lose a life. (This part of the game was more or less appropriated from a tabletop shoot-em-up which me and the Baughurst Piano Wizard used to play in the Hind's Head pub in Aldermaston Village). Meanwhile large rats would appear at the top of the screen and scurry downwards; you had to try and hit them with your hammer, killing them. If they got to the bottom of the screen they would chew through the floor and the pointy stick guys would get out and kill you.

The game didn't look too bad by the standards of the day, and featured a somewhat Monty Pythonesque death scene where a long arm would appear from the top of the screen to snatch away an unsuccessful player; but it just didn't play very well, no matter how much I tweaked it. In the end it was only on sale for a month or two before I decided to drop it from the Llamasoft lineup as simply not up to scratch.

I had a little more work to do with Defenda, too. In our discussions with the guy from HES, it became apparent to us that it really wasn't safe to just be barefacedly selling a version of Defender, and even calling it “Defender” with a slightly different spelling. The mighty Atari actually owned the rights to produce Defender for game consoles and home computers, and the word was that they were shortly going to get heavy with any cheeky people who thought they could get away with selling their own versions. To this end, for the US release I would have to change the name of the game and alter the graphics, and since Llamasoft was now starting to appear as a significant blip on the radar of the games biz in the UK, I figured I should do the same for our own version of the game for the home market.

To distance ourselves a bit from “true” Defender, I changed the graphics of the enemy ships so they didn't look like the Defender ones, and – since it was a Llamasoft game, after all – I replaced all the Humanoids with llamas. Since the game now featured flying through mountains in a skidpan-bus to rescue llamas that were being attacked by alien enemies, it seemed natural enough to rename the game as “Andes Attack”.

That was certainly a busy summer... and as it drew to a close, the strained relationship between ourselves and the Smegheads finally reached breaking point. My dad recalls the sequence of events:

It was at about this point, in July, that strains began to show in the short-lived partnership which had been established between ourselves, and Smeghead and Smeghead Sr. The Smegheads clearly wanted to run the show with the aim of making the maximum financial gain and to hell with the quality of the product, Jeff saw Llamasoft as the vehicle through which he could develop and publish his ideas and which, incidentally, would produce that useful byproduct, cash. These contrasting ambitions could not coexist, given the characters of the principal partners and relations became progressively more uncomfortable. In the meantime sales of the games were increasing steadily, as was Jeff's reputation, evidenced by the fact that in July he was interviewed by "Computer and Video Games" magazine.

In August I recall attending a ZX Microfair show in London, with Jeff and Smeghead, at one of the Horticultural halls near Victoria. We had two trestle tables, covered with army bedspreads supplied by Smeghead Sr., on which to display our wares, with our posters flyposted on adjacent walls and columns. This was my first experience of the Show atmosphere. We all seemed to enjoy it in spite of the interpersonal tensions.

On 16th August there was the first of a series of meetings between ourselves and the Smegheads to discuss the terms for the dissolution of the partnership (Somebody remarked that "A partnership is the only ship that doesn't sail anywhere."). The Smegheads were obstructive, the main issues being the ownership of the copyright of the games and the division of Llamasoft equipment and financial assets. There was also the question of the fate of the name "Llamasoft" which now had a certain prestige. The agreed outcomes were that the partnership would end,

that we would retain the Llamasoft name and the copyright of all Jeff's existing games.

Profits arising from their future sales would be split equally between the two factions, as would the contents of the partnership bank account.

The Smegheads were very reluctant to sign cheques releasing our share of the money (probably less than a thousand pounds) from the existing joint account hoping I am sure, that this would put us in financial difficulty. They had the backing of Smeghead Sr.'s other business and were in a better cashflow position than we were. The money was finally released at the end of 1982.

The last joint action by the partners was attendance at the Personal Computer World Show at the Barbican in London. I remember it as a few days of uncomfortable tension. Smeghead was like the spectre at the feast and I felt that he spent much of the time wondering how he could make life uncomfortable for us when the Partnership was dissolved at the end of the show.

Separation from the Smegheads formally took place on September 19th 1982.

More programming work also went on in the last couple of months of that Summer. Llamasoft needed more games for the unexpanded Vic – the 8K games were fine, but not everybody had the 8K memory expansion, and we needed some games that everybody could buy. The first of these was a relatively simple “bottom-shooter” game that comprised elements from a number of common shooting games of the time. You had several waves of attackers who would stream across the screen on predetermined flightpaths, and then at a point spread apart and descend to attempt to abduct little guys from the

bottom of the screen. You had to attempt to shoot down the ships before they got to your little guys, or if you were bold, you could actually allow the guys to be abducted and then shoot the abducting ship as it made its way up the screen, earning a bonus for rescue. You started out with a single, small ship that fired one bullet at a time, but if you survived long enough you'd be rewarded with a larger ship that could fire three missiles at once. The game itself was pretty simple, but it was actually quite fun and frantic to play, and the sound effects were pleasingly noisy and zappy. Given the abduction/rescue theme of the game we called this one "Abductor".

I also got a switchable memory cartridge for my Vic, which allowed me to simulate a ROM cartridge by placing some RAM at the location where normally the ROM would go. This allowed me to start work in preparing the US version of Andes Attack, which was to have further modifications to distance itself even further from Atari's Defender, and was to be called "Aggressor". It took me a while to get things properly sorted for execution out of ROM (I had a bad habit in those days of just allocating myself bits of storage any old where in RAM, sometimes right amongst the code itself, and since all that code was going into ROM, I needed to make sure that all my storage was really in RAM and all my code was really in ROM). Once I had a bootable game image that worked out of my simulated cartridge, I went to Reading and borrowed the facilities of a software house there called Audiogenic, who were themselves producing Vic-20 cartridge games, and who had an EPROM burner and cartridge PCBs that I could use to make an actual cartridge to test the game. Eventually I had code that worked out of ROM and which was non-Defenderised enough for HES

to release in the US, and that was sent out to Jay Balakrishnan in California.

In September it was time for us to go to the biggest computer show we'd yet attended, the Personal Computer World show at the Barbican centre in London. As my dad mentioned above, this was to be the last show that we'd attend as Llamasoft-plus-Smegheads. The show itself I recall as being stupidly, monstrously busy. It was four or five days long, and every single day it was crammed to bursting. It was very hot, cramped and uncomfortable in there, and the endless din of game sound effects and tunes (our own and countless others) was quite wearing; my mum was in attendance too by now, dealing with the business of taking payment for the games and providing nice cups of tea for all at appropriate times. Every night we left the show exhausted. I assumed what was to be my major role at shows to come, which consisted largely of hanging out in front of the Llamasoft stand, talking to the players (by then we even had a few who could be called "fans" of Llamasoft), demonstrating and playing the games.

A couple more significant events occurred at that Barbican show. Jay Balakrishnan was there again, and on one of the days he extracted me from the show for an hour or so and took me to the hotel where he was staying, where he presented me with an American version of Commodore's latest machine, not yet out in the UK – the Commodore 64. The only other one I'd ever seen at that point was another American one that Rabbit Software had on their stand at the show, so I was quite chuffed to receive this much-anticipated new machine. I couldn't actually use it straight away, since it required an NTSC display and a step-down transformer to run, neither of which I

had, but I could always acquire those in due course on a future trip to Tottenham Court Road.

After each day's show was over we'd sometimes head off into London to eat or to go to the arcades in the West End, and at that time the excellent movie "Blade Runner" was just out in cinemas and the walls of the Tube had many posters up advertising it. I hadn't yet seen the movie myself (I would get to enjoy that a couple of weeks later), but the posters were striking; the name of the film and the typeface of the font looked interesting... and for some reason I was thinking about the next game I wanted to do, which I wanted to be another unexpanded Vic game but wasn't sure what it would be like at that point. And while staring at one of the Blade Runner posters waiting for the Tube, the name "Grid Runner" popped into my head. That felt like it could be a cool name for a game. Now all I had to do was think of a game to fit the name.

Finally this exhausting show was over, and we headed off back home tired and a bit stressed by the whole Smeghead situation, which was coming to its inevitable and somewhat graceless conclusion. It all finished at my parents' place immediately upon arrival back from the show. The atmosphere wasn't exactly cordial, but agreements were agreed, the terms of separation defined, and that chapter of the Llamasoft era ended with Smeghead carrying the small colour TV that I'd used for coding on out of my bedroom, since it belonged to them.

That was not the last that we'd see of the Smegheads. However, it was an absolute relief to be shot of them and for Llamasoft to be fully ours at last. Show-weary and stressed, we drew a line under

the whole period by ordering a particularly large curry from the Indian at the end of the road and devouring it.

In the days following I thought more about what our new unexpanded Vic game should be like. Another arcade game that was popular for being cloned on home computers was Centipede, and I quite fancied doing something Centipede-like, especially since I'd already done a moderately successful Centipede game on the ZX81. However, Atari also owned the rights to Centipede, and with their recent sabre-rattling about illicit clones it seemed ill-advised to do just another Centipede clone. Besides, I needed something that would fit the name I'd thought of, "Grid Runner", so the whole garden theme would be out anyway. It needed to be a bit more spacy, a bit more hard-edged, a space shooter rather than a garden-based pest-em-up.

I began on a Sunday, I recall, just after "Battlestar Galactica" finished on telly. I was having to use the family telly again, since the small portable the Smegheads had loaned me had gone back with them. I sat on the floor in front of my Vic and used my character editor to design a font that gave me the Blade Runnery feel I wanted when you spelt out "GRID RUNNER". I created characters that could be repeated to make up the Grid which would comprise the play area. I made a little green ship, and a white bullet.

Over the next couple of days I sat in front of the Vic, on the floor, in front of the family telly, gradually putting together small bits of code into the shape of a game. First a simple routine to generate the playfield, and put the game's name and score at the top of the screen. Then some code to place the player's ship on the grid, and to make it

move when you moved the joystick. Then some more code to make bullets come out of the ship when you pressed the FIRE button, and travel up to the top screen edge.

On Wednesday nights there was a small club of enthusiasts who would meet in the local computer shop in Basingstoke, and I went to the meeting in midweek with a tape of my work in progress which I thought they might be interested to see. By that time there were some snakelike aliens which made their way back and forth across the grid, and you could move your ship and fire at them, but you couldn't yet actually shoot them. The people at the club seemed to quite like the way it was shaping up.

During that week me and my dad also made a trip up to London, to go to Tottenham Court Road to acquire the necessary bits of equipment that I needed in order to be able to use the American Commodore 64. We returned with a multistandard JVC monitor, quite a posh bit of kit in those days, and a stepdown transformer. This also solved the problem of my not having a telly to work with in my bedroom, and I was able to carry on with the development of Gridrunner at my desk in my bedroom instead of on the floor in front of the telly.

Gridrunner had to wait for a few hours the first night I got the Commodore 64 working – I wanted to at least have a look at some of the new functions of the new machine, so I spent that first evening in my room with the C64 manual and wrote a simple version of my old game “Rox”, all in BASIC but at least using a small part of the new sprite graphics and extended sound chip of the new machine.

However, the Vic-20 game was the first priority, since the Commodore 64 wasn't even on sale in the UK yet, so the following days I got back to the work of finishing Grid Runner.

Although we had the Centipede-style enemies, I didn't want to have mushrooms and spiders, since I was aiming for a different feel in Grid Runner. Without mushrooms, however, the trajectory of the snakes on the playfield was too uniform and boring. We needed some equivalent of the mushrooms to make their motions more interesting. I chose to make it so that when you shot a bit of the “centipede”, it'd leave an obstruction on the grid which would cause any centipede encountering it to turn and descend a level, like the mushrooms did in Centipede. This way the actions of the player would themselves generate obstacles on the playfield that would give the snake-enemies more interesting paths.

Like in Centipede, I made it so that the player could choose to shoot away these obstructions, with successive shots reducing them in size until eventually they shrank to nothing and disappeared. But unlike in Centipede, where apart from being obstructions the mushrooms themselves are harmless to the player, I built in an extra threat to mine. I turned them all into time-bombs. If you didn't shoot them away immediately, then every few seconds the “Pods” left behind by shooting snakes would evolve through one step, getting larger. Eventually, every unshot Pod that had been left on the playfield and allowed to “mature” would turn into a bomb, which would streak down the grid, destroying the player if he was caught unawares.

The final twist that I added was to add a couple of guns, one on the bottom edge of the grid and one at the left edge. These moved constantly and, at a regular interval, would fire lightning bolts which would instantaneously kill the player if he got caught in them. Where the two bolts met a new Pod would also be generated.

I added a deep, threatening “zap” sound effect that triggered whenever the guns fired and the Pods evolved, and this gave the game a satisfying, sinister “heartbeat”. The extra tension imparted to the game by these modifications seemed to work quite well, with the player constantly having to keep an eye on the position of the guns and the state of the Pods, and make sure he wasn't in the wrong place when the lightning bolts fired and the Pods matured.

As the weekend came around again I was finishing off the details of the game, separating the gameplay up into twenty successively difficult levels, making a transition screen between one level and the next, writing code that kept track of the score and the number of lives left and declaring Game Over at the appropriate time. In the end I was very tight indeed in the small memory space of the unexpanded Vic, and I couldn't even put my full name on the title screen, having instead to shorten my signature to my three initials “JCM”. Grid Runner became “Gridrunner”, too, because there wasn't much room on the top line of the Vic display to put the game name, player's score, and lives left indication.

It was all finished by the following Sunday night, just in time for me to relax and watch the next episode of Battlestar Galactica. I was pretty pleased with the game, since we were needing more stuff for

the unexpanded Vic. Jay Balakrishnan had asked to be sent copies of any new games I did, just in case any more came out which he might want to publish through HES, so the following week I sent a tape of the new game out to California. To be honest I didn't think much about that side of things, because he hadn't been interested in Abductor or Traxx, and I didn't really think he was into any of the new games very much, particularly the smaller, less complex unexpanded Vic games.

As it turned out, I was wrong about that.

Part 9: Mutant Camels

With Gridrunner finished, we prepared what was actually a very low-key launch for the game; if I remember correctly it was just quietly added to the list of available VIC games – at that time we had no reason to expect that it would be anything more special than just another unexpanded VIC game. I had the shiny new C64 to look at as well, so having finished up Gridrunner I set out to create my first proper game on the new Commodore (I don't really count Rox-64, since that was really just an evening's warmup to get a feel of how the thing worked).

Now, I was in the habit (as all geek kids in those days were) of going into computer and videogame shops and playing on all the demo

games, regardless of intent to actually buy any; it was simply a good excuse to sample what was available. This time was pretty much the heyday (and, as we would find out the following year, the last hurrah) of the Atari VCS, and the games for Atari's little machine were everywhere. Upon one occasion I'd been with my parents down in Winchester for some reason, and had sloped off to find a computer shop and have myself a helping of free gaming, and there I'd come across Parker Brothers' "Empire Strikes Back" game for the VCS, which I played for a while and quite liked. It featured horizontal scrolling (and even a primitive scanner) like "Defender", but instead of lots of small enemies you had to fight an ever-advancing column of lumbering Imperial Walkers, circling round them and blasting them with shots repeatedly until they succumbed (the humble VCS was far from being able to implement any malarkey with tow cables or anything). If you destroyed them all then you'd go through a short warp sequence and then the next level would begin (exactly the same as the previous level but a bit faster and more difficult, which was the way of games in those days).

I quite enjoyed that game, but I didn't really think much more about it once I was home from Winchester, and certainly it never entered my head that I might want to do any kind of a copy of it. However... well, I think the blame lies firmly with Computer & Video Games magazine, actually. I happened to be flicking through C&VG one day and spotted a review of the Parker Brothers VCS game. Having played it myself and quite liked it I was interested to see what the reviewers had thought of it, so I had a quick read through of their review.

I can't even remember if they liked it or not. What I do remember, however, is that they described the lumbering Imperial walkers as being like “giant, mechanical camels”.

Now, I like camels.

I like camels a lot. (They do, after all, share common ancestry with the camelids of South America, of which the llama is an example). And therefore you can blame C&VG for the fact that when I was considering what to do first on my new Commodore 64, images of giant camels came to mind. Giant, laser-spitting camels. Giant, laser-spitting, neutronium-shielded mutant camels, to be precise.

I wanted to do some kind of scrolling game for my first C64 outing, and thanks to that helpful nudge in the right direction from C&VG, I thought I'd do a nice version of that Parker Brothers game I'd seen in Winchester, except done with 90-foot-high, neutronium-shielded, laser-spitting Mutant Camels of Doom. And so I set to work on the game I called “Attack of the Mutant Camels”, or AMC for short.

I seem to remember my parents being away at the start of this work; I think they'd gone off to America, since my dad's job at the Bomb Factory occasionally sent him off to places like the Lawrence Livermore labs, and so a lot of the early work was done on my own, in the traditional bedroom, banging out the code and learning my way round the new machine. Defining fonts was easy, exactly the same technique as on the Vic-20, so I wrote myself a little character editor and designed a cool-looking font for the game. There were these new sprite things too, so I wrote myself an editor for those too, and set about

defining the sprites for the game. I made bullets, a little spaceship, and other bits of screen furniture like an AMC logo and a scanner reticule. As for the camels, I wanted them to be really large on the screen, so rather than use single sprites (which, even in “expanded mode” didn't fill enough of the screen to be imposing enough) I decided to use 4 of them all bolted together to make one huge camel sprite. It must be said that these large sprites severely overtaxed my meagre artistic abilities, and to be honest my Mutant Camels ended up looking like two fat men in a pantomime Mutant Camel suit. Despite this I chucked them in the code and before long had a landscape around which I could fly (in a somewhat chunky fashion, since I had yet to unlock the mysteries of raster interrupts and the scroll registers) and which was populated by a line of lumbering pantomime Mutant Camels, just like in the Parker Brothers game, but considerably sillier.

About halfway through the development of AMC, however, there was an interruption which caused its development to be put on hold for a couple of weeks – an interruption which became part of Llamasoft historical lore.

One morning, very early, probably at around 4am or so, I was disturbed from my dreams of Mutant Camels by the phone going. I emerged from my pit and staggered out to answer it, and it turned out to be none other than Jay Balakrishnan, calling from California, where he'd just received the copy of “Gridrunner” for the Vic that I'd sent out a week or so previously. Jay sounded very happy, and was enthusiastically telling me how they'd just been sitting for hours playing Gridrunner. He was of the opinion that it would sell rather well, and that I would therefore be in line to maybe receive some good

royalty payments, and he's just called me up to be all enthusiastic about it and let me know how pleased he was with the game. Which was excellent to hear, and I wrote it all down on a bit of paper next to the phone and then, as it was 4am, staggered off back to my pit and went back to sleep.

Later – much later – that morning I emerged once more from my pit at the usual time and staggered out into the kitchen to make a nice cup of tea. (Without tea there would be no Llamasoft, no coding and, indeed, no British software business at all. The lines of code flow out upon a river of tea; this marvellous substance lubricates and stimulates the synapses and helps them to fire in a manner conducive to the generation of practical and robust code. Tea begins and punctuates the day; whenever a moment's pause for thought is needed it is to the kettle that we turn, that humble appliance, ubiquitous in every British home, without which the wheels of industry would cease to turn. Many a cool idea or neat game embellishment began life in a head which had just recently taken a refreshing gulp of piping-hot PG).

I was booting the wetware to the operational levels needed to begin the day's coding by consuming my morning cuppa and my progressively sharpening synapses recalled the incident of the phone call at stupid o'clock. At first I thought it might have been something I dreamed (after all, having a dream about someone calling you to say that your work was ace and that you were going to make good money seemed entirely feasible, more so than it actually happening in real life), but I went to the phone and there were the notes I'd dozily made which appeared to confirm that it had actually happened.

This was further confirmed when, later in the day I received further contact from Jay, who reiterated that he thought Gridrunner was

ace, and informed me that he would like a version prepared for the new Commodore 64 as well, and as soon as possible.

This meant that I had to set aside my camels for the moment, and instead work on producing Gridrunner for the Commodore 64. This actually wasn't that difficult to achieve, since the Commodore 64 and the Vic-20 shared the same 6502 CPU, which meant that the bulk of the game logic would still work. The size of the screen was larger on the Commodore 64 – 40 characters wide instead of 22 like on the Vic, and the code needed adjusting for that. The sound code also needed redoing because the SID sound chip on the c64 was quite different (and more powerful) than the sound generator of the Vic-20.

Given these similarities it was only a couple of days before I had a working version of Gridrunner on the C64. To be brutally honest I don't think it looked or played as nicely as the original Vic-20 version – the graphics looked a bit thin and weedy compared to the bold, fat graphics imposed on the Vic by its lower resolution graphics system. If I'd taken more time to adapt it to its new home, and make more use of the C64's more powerful hardware, I could have made it a much better game; as it is it's OK but the Vic version is better in my opinion. However HES wanted a version very quickly, so I basically just did the port and then prepared a ROMable version and that was it, necessitating only a few days' break from the AMC project. It also gave us another game that we could sell, our first for the new C64. But despite Jay's phonecall I still don't think I really thought Gridrunner would do so exceptionally well, and after the new version was shipped off to Jay I got back to the real work, my pantomime camels game.



“Attack of the Mutant Camels” for the Commodore 64.

AMC was quite a simple game and I finished it off in another week or two, adding in player shooting, camels shooting, collision detection, lives, score and such, all the usual furniture of a simple shooting game. A hyperspace sequence was added for the transition between levels, a hi-score feature implemented, a title screen where the “AMC” logo formed up out of sprites, and I was done. This was to me the first proper C64 game, since it at least attempted to use some of the new Commodore's extra hardware and was actually started before Gridrunner for the C64.

Once done I again sent a copy of the game out to Jay in California, and I confidently expected him to like it rather more than the (to my mind) slightly disappointing port of Gridrunner that I'd sent previously.

Throughout all this we continued with other parts of necessary Llamasoft life, much of which at that time involved going to computer shows in London where, just as at that first Commodore show in the Novotel, we would basically just put out a bunch of machines on a table, load in the games, and let people play them – this was pretty much the extent of our marketing efforts.

My dad recalls the typical proceedings of the time:

Early in January 1983 we had a stand at the London Home Computer Show, possibly the first we had attended since the end of the partnership with the Smegheads. It was in one of the Horticultural Halls; we would have had all our products on show- for the VIC 20, CBM 64 and Spectrum machines. This was also a period in which we were learning about and refining the techniques, tricks and wheezes associated with these events; transporting, parking and setting up; coping with the inevitable minor crises (missing cables, not enough power sockets, mistakes on the display posters, etc etc), dealing with the customers, a minority of whom were plausible and devious, being alert to the possibility of theft of joysticks for instance, and finally organising the dismantling and packing of all the equipment as smartly as possible at the end of the event.

It was all quite hard work and the pressures at times were high but there were rewards- primarily the feeling that most of the customers appreciated the value and the quality of the games and came back to us again and again. In fact as the months went by we accumulated a loyal band of Llamasoft “groupies” who would turn up to every show at which we exhibited and assist in various ways, especially in the dismantling phase. They clearly enjoyed the contact with Llamasoft and with Jeff in particular. Particular friends were Wulf Grimbly, Marco, and Matthew Spall.

I'm not sure “groupies” is quite the appropriate word, but certainly a nice little bit of a social scene started to take place at the shows, and there was an excellent little core of Llamasofties that developed who would always be on hand to help out around the stand and help us to tear down at the end, which was much appreciated, because by the time these shows finished we'd always be extremely tired and want nothing more than just to get back home to Tadley and order a large curry. Since the shows were usually multi-day affairs we'd often take hotel rooms in London for the duration rather than thrash up and down the M4 each day, and this allowed for further socialising with the 'softies in the evenings. We'd usually all set off on the Tube somewhere, as often as not to go and hang out in the arcades around Soho and Leicester Square where there were often good games to be found. In later years trips to the London Planetarium to catch the Pink Floyd laser show were also a regular occurrence. I had some great times at those shows over the years, and got to meet some great people and enjoy good friendships with them. That kind of computer show didn't really survive into the 90s, though, and the “trade shows” of today are an entirely different animal. One of the things I must miss about “the old days” is the charm, enthusiasm, friendship and fun that went along with the whole show experience, tiring though the shows inevitably were.

Dad also remembers, at that time:

Jeff's next game was also well advanced- LASERZONE. LZ featured one of the innovative control systems which have been typical of Jeff's work; the player simultaneously controlled the movements and firing of two laser weapons which travelled along tracks which formed one of the

vertical and one of the horizontal edges of the screen. The alien creatures who were the game threats appeared within the area so encompassed and had to be dealt with before they they “landed” on the appropriate axis and destroyed the laser bases. The game was designed for the CBM 64 and translated for the VIC 20.

(Actually he has it a little wrong there; the game was definitely made for the Vic first, and the C64 version was made after. Although the C64 was a newer and more powerful machine we didn't simply abandon the Vic-20 overnight. Indeed, at first the new Commodore was prohibitively expensive for many people (I think it was around £400 when it was launched) and the little Vic remained our primary market for many months to come, and so for a while the Vic remained my primary focus with the C64 receiving conversions after the Vic versions were done).

Laser Zone was an attempt to try a new game style – although obviously still not quite above borrowing “inspiration” from other games as I blatantly had with AMC and that Parker Brothers game, I was also interested in trying out new stuff and coming up with my own ideas. Laser Zone took the idea of the classic “bottom shooter” game (where, as in Space Invaders, the player ship is at the bottom of the screen and shoots upwards at enemies coming down from above), rotated it by 90 degrees, and overlaid it on top of itself. So you had two shooters, one at the bottom of the screen and one at the side, and you controlled them both at once with the joystick. Enemies appeared both at the top and at the opposite side, and you could use either gun to shoot at them. If an enemy landed on an axis where one of the guns was, it was possible for a skilful player to shoot over from the other

axis to remove the threat. Laser Zone was considerably trickier to play than Gridrunner, but nonetheless I felt that it worked reasonably well, and I was particularly happy with the sound effects – I'd been able to get some really nice, solid-sounding effects out of the Vic sound chip, almost comparable to the sound effects on Williams' awesome Defender coin-op that I greatly revered.

At about this time – I don't recall the exact timing of it, but it was certainly at around the time of completion of Laser Zone – we finally started to get some money from the deal that we'd done with HES for selling Gridrunner in the US, and just as Jay had told us, it looked like being a fair bit more than we were used to making selling the games ourselves at home. They had been happy to receive the Commodore 64 port of Gridrunner and another deal was struck for the distribution of that, but (curiously, I thought) they didn't seem to like my “Attack of the Mutant Camels” for C64. I wasn't unduly bothered by that since they seemed so happy with Gridrunner and since the Vic was still a large market for them just as it was for us, I sent them the completed “Laser Zone” for evaluation.

Having completed Laser Zone, both me and my mum decided we needed a break. Things had been getting increasingly hectic since Llamasoft had started only a few months before, and between all the shows and the coding we were both pretty knackered and in need of a rest, so in February of 1983 we both buggered off to chill for a week in a hotel in the Algarve. I was still not fully up to strength after my period of illness and according to my mum's memory I actually slept uninterrupted for two days upon arrival at the hotel. I remember having a nice chilled-out time, buying foreign computer magazines

from the newsagent kiosk in Albufeira, and reading “Dune” whilst lying on the beach (it was February and therefore not swelteringly hot, but nice enough compared to a British winter). I bought a jumper with a large blue camel on it, and we went on several touristy excursions including one up a mountain, where it proved to be bloody cold and in fact snowed, much to the surprise of the locals.

I also distinguished myself in the local arcade, to which I would slip off for an occasional helping of arcade action throughout our stay. They had a Phoenix machine there, and it must have been a fairly recent arrival, because the locals plainly didn't know how to extract the maximum amount of bonus points from the mothership. I went to play and, as any fool knows, I didn't shoot the mothership at the first opportunity, instead choosing sensibly to let it get as low as possible before essaying the finishing shot. At first this caused the local lads to attempt vociferously to inform me that I should shoot the ship when I appeared to them to be pointlessly idling underneath its descending bulk, but their confusion soon turned to respect when I finally (somewhat smugly) shot the little bug guy in the centre to receive a whopping 8,200 point bonus and an extra life.

Part 10: Go West, Young Man

Back home from holiday (and how I miss having holidays!) and straight back into things, judging by my dad's notes here:

On 26th of February Hazel, Jeff and I were on the Llamasoft stand at the Commodore Show which was held in the Westminster Central Hall. AMC and LZ were our main attractions. No doubt, the Smegheads, in their new guise as “Glaring Grammatical Error Software” were at this show. We encountered them at virtually every show for the next eight years or so. This was not really a problem, we simply kept our distance from them. They tried to needle us by , for instance, copying a character set which Jeff had designed into one of their games (but then so did others..) and by having having Llamaburgers feature in one of them!

The character set stuff was the least of it, really – for some reason the little character set I'd done for *Attack of the Mutant Camels* (and later refined for its sequel, *Revenge of the Mutant Camels*) seemed irresistably popular to many game developers of the day, who had no compunction at all about helping themselves to it and using it in their own games. It irked me somewhat, but not a huge amount; I just thought it was a strange thing to do, because for me part of the initial phase of creating a game was sitting down with the character editor and coming up with a font to suit the game. It set the stage for everything that followed, and helped give a game its identity. Simply blagging someone else's seemed to me to be omitting a gentle, necessary part of the process of making a game, somehow.

Glaring Grammatical Error Software did seem to have the proper hump about our split with them, and the fact that we seemed to be doing rather well afterwards, since they couldn't resist having snide little digs at us in just about every bit of software they released for a while. Every game they did would have direct or indirect derogatory references to me, to Llamasoft, and to the animals which became our theme. This was more laughable than genuinely annoying, and in the end they gave themselves enough rope and started to get torn apart in reviews where even the reviewers found the constant cheap shots unnecessary and annoying; once they started getting bad press about it they pretty soon stopped it, although I can't say that relations between us were ever cordial.

The next month I was off on my travels again, as my dad notes:

In March Jeff went to San Francisco for the first time, to visit HES, for about two weeks. He was well received- I remember that he said he was introduced to someone by Jay Balakrishnan as “the best VIC 20 programmer in the world” ! He also had contact with other programmers who worked directly for HES and I am sure gained much from that experience. And of course he took in the sights and spent time and money in the games arcades on Pier 39.

I think Jay was certainly being a little overenthusiastic there – such advantage as I had came purely from having a couple of years' more experience at making games than most people, thanks to my experience on the PET at 6th form. I certainly wasn't technically any great shakes; I think I was lucky in that people liked the games for what they were rather than for any great degree of technical excellence. I was a competent coder but by no means a genius at it. If it was fun to play and didn't fall over then I figured I'd done well.

In fact one lad I met on that trip to the US, a chap called Tom Griner, had produced some of the best Vic-20 code I'd ever seen. His work for HES used similar trademark effects to the Williams Electronics stuff I so admired in Defender – proportional fonts, nice bitmapped explosions, simulated software sprites – all things much more advanced than the purely character-based stuff I'd done for them.

That first trip to the US was quite a landmark for me, though. For one thing it was my very first trip to America, and it was all paid for by HES. I remember it all felt somewhat unbelievable to me. I cued up “Countdown” by Rush in my Amstrad Walkman so that it would come on just as the 747 was accelerating down the runway, and,

once airborne, just sat and had a bit of a think about all that had happened in such a short space of time. Not much more than a year previously I'd failed out of University and then even had to bail out of a polytechnic course at Oxford when I became unexpectedly ill, and for a while I hadn't known what the hell I was supposed to do next and just felt like a complete failure. And yet here was my arse on a plane, flying to California with a G&T in my hand, apparently in demand and on course for a career in something which didn't even exist during the time that I was passing through the educational system. I didn't have a single qualification to my name in what I was actually doing, and nothing and nobody – no exam or careers advisor – could possibly have prepared me for it. It all seemed very strange.

The plane stopped over briefly in LA on the way in, and I remember tuning in to the local radio stations as we sat on the runway, intrigued to be tapping in first-hand to American culture through the radio – I'd only ever seen America on films and the telly and this was a long way from Tadley and all my life that had gone before. We'd flown a polar route over and just seeing the sheer amount of ground that went by as you came down over Canada and the US was amazing – fly back into the UK and by the time you get over land, you're already in the descent to Heathrow. Continental North America went on for hours. It was quite an eye-opener, especially after a few G&Ts.

Upon arrival at San Francisco I was met by Jay, who drove me back to the HES headquarters near Berkeley where they had not long previously taken delivery of a coin-op Pole Position machine. Again, this was amazing, since the idea that people could buy these machines and just have them set up on free play to have fun on rather than having to pay for each game at the arcade had never really occurred to me.

That first night I spent having friendly competitions with Jay and the others at HES on the new machine, although I wasn't up too late because I was beginning to understand what jetlag felt like. In the end Jay took me back into San Francisco and the motel that they'd booked me into for my stay. It wasn't posh, just a typical American motel on the corner of Van Ness called the Oasis (no camels though). I didn't care though, it was excellent to me precisely because it was a typical American motel, just like I'd seen on telly, and I could wander round nearby bits of the city at will, absorbing American culture and Kentucky Fried Chicken.

The reason for this first visit was, if I remember correctly, to attend the West Coast Computer Faire that was held in downtown San Francisco once a year at that time. Although I remember that being fairly interesting (and being amazed by the likes of “Wayout” on the 8-bit Atari, a first-person maze game that in a way prefigured the likes of “Wolfenstein 3D” and, more distantly, “Doom” which were to come along many years later), what I most remember about the trip is the time I spent simply walking around San Francisco and exploring, both spatially and culturally. Since it was my first ever trip to the US I had decided that there were certain things I had to try, things that were culturally iconic of America.

I sought out a diner and got myself what I considered to be an American breakfast. I know that hash browns were involved, and most definitely maple syrup and pancakes – which I hated. Far too sweet and sickly to be having at breakfast, if at all!

I went to McDonalds and got myself a burger and fries and root beer. Always we'd hear about root beer in American culture, surely it had to be something good... but yuk! It tasted like dentists' mouthwash! Vile stuff, how could anyone possibly drink that for pleasure?

It wasn't all disappointment, of course. I managed to find my way to a good steakhouse and verify at first hand that the Americans did have a most excellent way with dead cows (even if we shall, as a matter of decency, leave the matter of bacon unmentioned). There was much to enjoy in this vibrant foreign culture that I found myself so unexpectedly immersed in, those sunny Spring days in California in 1983.

I did quite a lot of walking around in San Francisco on that visit, mainly for one reason – the new and revolutionary Vectrex console had just been launched in the US, and I was determined to find one and bring it home with me to the UK where it wasn't yet out. It wasn't long out over there when I visited, however, and supplies were very short indeed. I trekked all over San Francisco trying to find one without success, until eventually I found a store which had a couple left. I don't remember which store, but I do remember it was a good few miles from my motel, and I had to walk back carrying a large plastic sack containing the Vectrex and a few games. Well worth the effort, though, and the rest of my stay was livened by many games of Mine Storm, Berzerk, Scramble and Star Trek on the Vectrex in my motel room.

I explored the more touristy end of town, down by Pier 39, primarily in search of the good arcades which were common back then. There was a single, massive arcade on Pier 39 itself which had an entire room dedicated to Pac-Man and rank upon rank of popular games such as Space Invaders, Galaxian, Asteroids and of course Defender. There were various other arcades scattered around in that general area and one could quite happily spend all day there bouncing from one to the next with pockets full of quarters. Games I recall seeing there for the first time include Quantum, Time Pilot, Reactor, Tac-Scan, Star Trek, Zoo Keeper (I was highly amused at the level with the tiny, manic camels bouncing around all over the screen)... I also listened to the radio a fair bit and was quite enamoured of some of the stations there, in particular KMEL 106, whose logo was a gormless-looking flying camel and whose DJs always sounded like they were comfortably stoned and played lots of Floyd, Yes, Genesis, Marillion and such – exactly the kind of stuff I listened to myself at home. In fact the first official airing of the Floyd's new album “The Final Cut” occurred while I was out there, and KMEL made a big deal of the lead-up to this momentous event, and I made sure to be listening when the album was finally played.

I spent a couple of days visiting HES's HQ near Berkeley – at the time they would very much have liked to have hired me into a full-time job with them, and it was a bit tempting; they had posh offices with adjacent gym facilities (including a jacuzzi in which, never having used one before and not knowing where the temperature control was, I contrived to boil myself nicely lobster pink one day). Intriguing though the offer was I didn't end up taking it – it would have been just too

much, I think, and it wasn't like things weren't moving fast enough at home, and that was where all my family and mates were, after all.

It was an excellent trip though, and I was well looked after by Jay and HES and left with very favourable impressions of American people and culture (root beer notwithstanding). I brought home with me various souvenirs and, of course, the coveted Vectrex. And some smokes for my brother – I didn't smoke at the time, and I think I had rather surprised Jay when he asked me what I was planning on taking back home with me and I'd replied “200 fags for my brother”.

Whilst out there it was clear that Gridrunner was indeed doing abundantly well in the US market, and one of the things I did whilst in my motel room was jot down some ideas on a bit of paper regarding a sequel. The original game had been coded for the unexpanded Vic-20, and had been a tight fit indeed, meaning that the gameplay had by necessity to be very simple and repetitive. I had it in mind to do a followup that used 8K of memory – although that was a much smaller market at home where only users with 8k or more expansion RAM could play it, it didn't matter so much in the US, given that for the US market the game would be in 8K of ROM instead – effectively the game was its own “memory expansion” so any Vic user could still play.

Accordingly I set about extending Gridrunner, adding a lot more variety as you played through the levels, more enemy types, new kinds of obstacles, better explosions – just generally improving on the design and making it less repetitive. There were hidden bonuses for the player to discover more more tricks that could be used for high scoring.

I returned home with the notes I'd made and set to coding, and by the end of April the sequel was done. I called it “Matrix”, and it was much more fun than Gridrunner – there were more grids with more enemies (and even tiny camels that ran down the grid and for which you got 106 points each, in homage to that nice stoner radio station that I'd enjoyed on my visit). The sound effects were nicer too, applying stuff that I'd learned when I'd made Laser Zone.

My dad describes the game:

By mid-April Jeff had produced another game, MATRIX. Matrix was derived from Gridrunner. Like the latter the player controlled a small spaceship was was able to move over part of the surface of a rectangular grid. The task was to destroy the snake-like alien creatures which travelled on the grid. Matrix added new features which significantly increased the complexity and the addictiveness of the game. For example it featured “The Snitch”, a humanoid creature who ran along the upper edge of the screen, pointing out to your enemies where you were! Gridrunner had been famous but Matrix was the better game, with more depth to it. Versions were produced for the 64 and the VIC 20 and sold well here and in the States.

As before I made a quick and dirty Commodore 64 port of the game, and copies were sent out to Jay for evaluation. The news from them was good – they wanted to take Laser Zone which I'd sent them a while before, plus both versions of Matrix – but they didn't like the name “Matrix”, but instead wanted to call the game “Attack of the Mutant Camels”! They didn't like the game of AMC but they liked the name, and they didn't like the name of Matrix but they did like the

game... and so it was that “Attack of the Mutant Camels” came to mean two completely different games depending on if you lived in the US or the rest of the world, and was doubtless slightly confusing to historians many years later compiling collections of emulator images of games from that era.

In fact HES were so fond of me at this point that they paid me a fairly considerable sum each month just for the rights of first refusal on anything that I came up with, a situation which was perfectly fine by me, since I lost nothing if they chose not to take a game as had happened with (my) AMC, and they were coming up with plenty of royalties on the games that they did take.

All in all, as my 21st birthday rolled around at the end of that April, it was a happy and exciting time for me and for Llamasoft.

Part 11: An English Country Garden

So on 22 April 1983 it was happy birthday to me, and I was 21. My parents got me a nice programmer's chair for my birthday, and that became my “coding throne” throughout the next few years. I also recall that for a birthday treat, they took me up to London to see a show, and while I was there another of those seminal events happened, something that at the time I didn't realise quite how significant it would be.

We got up to London pretty early – it wasn't that nice a day, overcast and occasionally drizzling. My parents split off to do some shopping and I went off on my own, agreeing to meet up later at Tottenham Court tube. I did my usual round-London mooching of the time, mainly hanging out in such places as the HMV and Virgin

Megastore on Oxford Street, with forays out to the tech shops on Tottenham Court Road (Lion House was a favourite, I think Silica Shop were based there, and you could go upstairs and pig out on free Atari for hours if you felt so inclined) and down to Piccadilly Circus for the Crystal Rooms arcade and many others which were in that area. I remember that I was walking down Shaftesbury Avenue and it had started to rain again, so I kept an eye out for somewhere to dive into and before long the entrance to a nice, dark arcade presented itself. I gratefully ducked inside and glanced around at the available machines as my eyes adjusted to the comfortable gloom. And I saw something beautiful.

There was a class of coin-op games back then which used a unique display method, completely different to how a regular telly works. Most games (then and now) are “raster” games. In a raster game, the electron beam scans the entire screen and lights up pixels on a grid (the grid is called a “raster”, and the raster has a finite number of pixels along each edge; in those days quite a low number – typically 320 across by maybe 240 tall). Game elements have to be built up of the pixels contained in the raster; those pixels are the display limit of the raster display, the “atoms” which cannot be subdivided. You can't display half a pixel, nor can you display motion of less than one pixel at a time. That's why raster games of that era look “blocky” - those primitive rasters had so few pixels that the ones making up the objects onscreen were visibly large, causing them to look like they were made of little Lego bricks. This low resolution, coupled with an ability typically to display only a very limited number of colours (often only 8 or 16) accounts for the primitive look of some of those early games.

However there were some games that escaped the limitations of the raster display by using a completely different display method, called “vector graphics”. In vector games the electron beam doesn't draw a whole grid of pixels. Instead it draws a series of thin, glowing lines (called “vectors”). Because the beam doesn't attempt to illuminate a whole grid of pixels, in a way vector displays are less capable than raster displays; you can't, for example, display scrolling backgrounds or, indeed, have much background that isn't black at all. However, their advantage in one particular area was considerable, and that was in their ability to draw true lines.

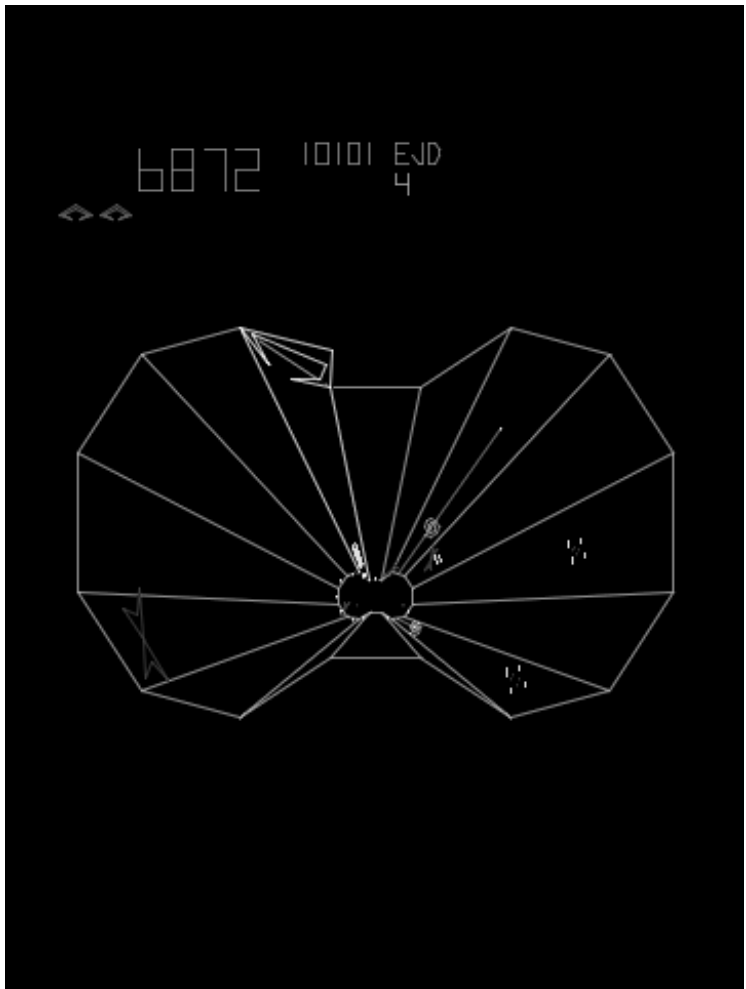
If you draw a line on a raster display, remember that you are limited to the size of the “atoms”, or individual pixels, in the raster. Which means that you can never truly draw a line; what you draw is a series of blocks, like a staircase. On those old displays the results for lines that weren't perfectly horizontal or vertical looked very ugly indeed, which is why raster games were usually made out of sprites and tiles rather than out of outlines. Even today, although the spatial and colour resolutions available are much greater, this stair-stepping of lines and polygon edges remains a problem – it's the cause of what gamers refer to as “jaggies”, and it's what the malarkey in your posh graphics card that deals with “oversampling” and “4x AA” is struggling to conceal.

In vector games, lines were perfect. This ancient, early-80s technology was able to display brilliant, beautiful, pure lines that even a modern system from the mid-2000s can't match. There were no atoms, no stair-steps, no “jaggies”; the electron beam drew the lines perfectly in beautiful, glowing precision. There were still huge limitations, of course – you could only draw a finite number of lines,

and that wasn't very many, so games based on vector displays tended to favour abstract, geometrical graphics styles rather than anything realistic. But the perfectness of the lines, and the sparse beauty of the delicate, glowing displays gave these games a unique beauty that in some ways has never been rivalled even by modern, monster GPUs.

Every gamer back then knew some vector games, and some were great classics, like Atari's awesome "Asteroids". There were many others – Star Castle, Lunar Lander, Battle Zone, Tail Gunner, Space Wars (itself a version of one of the first ever video games, created on a vector display on a PDP1 in 1961). The little console I'd bought in America, the "Vectrex", was the first and only attempt to bring these beautiful graphics to the home user (and remains just about the only way even today's gamers can play vector games at home, short of buying a coin-op from that era; the Vectrex is coveted to this day, and commands considerable prices on eBay).

Colour in vector displays was a relative novelty; most of the earlier games were in monochrome with colour provided if necessary by the old trick of coloured plastic layered over the screen. "Tac-Scan" which I'd played in the US was in colour; but colour vector games were rare, and just by dint of that distinctive enough to catch your eye in any game room.



Atari's “Tempest” coin-op.

What I saw in that dim arcade on that rainy day in London, though, was not just striking: it was beautiful. Bright, blue neon vectors picked out what looked like a three-dimensional spiders' web

floating in space. Intensely coloured geometric shapes “walked” on this surface, appearing down one end as tiny dots, growing in size and taking form as they moved up the Web towards the top. A bright yellow “claw” representing the player clung to the top of the Web, firing strings of shots that looked like tiny flowers down onto the enemies as they climbed up the Web. I don't think I'd ever seen a game, vector or raster, that looked so distinctive and so lovely. I gravitated straight towards the welcoming vector glow and put in my 10p.

RATE YOURSELF, the machine instructed me. I looked down at the controls and discovered another thing which helped give this game its character – instead of a joystick, you controlled the motions of the on-screen Claw with a heavy, weighted flywheel. This gave you a wonderful degree of precision in your motion – you could spin the wheel to fly round the Web at great speed but then just catch it to stop instantly, or move it gently to achieve the perfect position. The weight of the flywheel helped you to feel more connected to the game than if a joystick had been used. It was almost as if you could feel the mass of your Claw as you played.

Being a complete beginner I spun the wheel to select the simplest level, and started to play. The game put me at the top of a circular tunnel, and enemies started to climb up from the bottom. It was simple and intuitive to spin the controller and move my Claw around the edge, raining shots down on the enemies below. Before long I'd shot the last approaching enemy, and something wonderful happened. The Claw started to accelerate down the now-empty Web, accelerating into the screen and eventually leaving the bottom edge of the Web and zooming through a colourful 3D starfield, to the

accompaniment of a lovely, throaty warp sound effect. After a few seconds this warp noise and motion reached a crescendo and the words “SUPERZAPPER RECHARGE” appeared on the screen. The motion through the starfield started to slow and, off in the distance, a tiny blue dot grew and gained detail smoothly until you could see that it was in fact the entire next level, and you were flying towards it, eventually to land gently on the rim, ready for the next round.

Everything about this game – the pure, neon vector graphics, the wonderful 3D warp sequence, the excellent sound effects, the precise, comfortable, physical controller – everything was just wonderful. The gameplay was awesome too, as I spent a not inconsiderable amount of time and 10p coins finding out. There was even a fireworks display when you got your name on the hi-score table. It was beautiful. I'd finally found another game that ranked alongside Williams' awesome “Defender”.

Its name was “Tempest”. I had no idea back then, as I played in wonder in that dim arcade in London, how in later years I'd become associated with that game far more deeply than just as a player (nor, indeed, that one day I'd have a sheep named after the game, and a perfect arcade game of it in my own house to play whenever I liked).

But that day was the start of a big love which has never died.

Back home from the birthday treat and my Tempest epiphany, life for Llamasoft continued its hectic pace, and as my dad notes, we were off to yet another show:

Between the 28th April and Mayday we were in Birmingham at the Midland Computer Show, in the Bingley Hall. We now had a respectable range of VIC 20 and CBM games (the appeal of our Spectrum products was now reducing and all Jeff's efforts were directed to the Commodore machines). AMC, MATRIX and LASERZONE all attracted favourable comments and good sales.

In truth I'd never really taken to the Speccy – I'd done a few simple things on it just to dip my toe in the water, but I was far more enamoured of my beloved Commodores. For one thing the sound on the Speccy was weedy, coming from a simple beeper device rather than a proper synthesiser chip like in the Commodores. The colour modes were also a bit primitive and nowhere near as versatile as the Commodore displays, particularly that of the new Commodore 64 which offered arcade-machine-like capabilities such as sprites and smooth scrolling. I respect the Spectrum for the positive impact it had on the British computer scene, and its programmers pulled off some prodigious feats of programming to wring games out of it that nobody could have expected from such a comparatively primitive system; but it was not for me. I loved my Commodores.

We drove up to Birmingham in two cars (actually I think by then it was one car and a hire van). I'd only relatively recently learned to drive, and I had a brand-new but super boring Austin Maestro, in which I shared the journey with one of my mates from Tadley, Dave the Rock Star who I'd met at the Basingstoke computer club and who came with us to help out for the duration of the show. I suspect we

probably rocked out to some metal on the trip up, given the nature of my passenger. We weren't to be staying in Birmingham itself; rather, we'd arranged to stay at a place some miles south, in a more rural area, which proved to be excellent for reasons that will become apparent. Let my dad tell it:

A WARWICKSHIRE GARDEN

For the duration of that show we stayed, on a bed and breakfast basis, at a farm near Henley-in-Arden, south of Birmingham. The farmhouse was Elizabethan, occupied by a pleasant family which included two children who were quite fascinated by our computer games (Jeff spent all his leisure hours playing his own and other games). One morning as we all sat in the dining room enjoying our breakfast one of us commented on the farmhouse garden which was a well tended combination of close-mown lawn and high box hedges. By the time the meal had finished Jeff and I had devised the principal features for a new game; a lawn-mowing game in which an angry neighbour would pursue a houseowner who had borrowed his mower without permission. This idea was to become HOVER BOVVER.

It certainly was a lovely place to be staying in, way nicer than just staying in some hotel in the centre of Birmingham. It gave you some idea of what it must be like to be rich and posh and able to live in such a lovely old farmhouse, surrounded by beautiful gardens maintained by garden staff. It was more like staying at someone's place as a friend of the family rather than just being a hotel guest; we were well looked-after. I remember there being a lovely fireplace to sit

round in the evenings, and excellent cooked breakfasts each morning. In the evening Dave the Rock Star and I would slope off for a walk to the village pub, united by one of the sociable Walkmen you used to have back then that accepted two pairs of headphones, listening to Accept or some such bit of metallic goodness from Dave's collection, there to have a few pints each and stagger comfortably back, pausing occasionally to micturate as necessary in some handy rural hedgerow. It certainly beat staying in town.

It was whilst enjoying one of those excellent breakfasts one bright, sunny morning that Hover Bovver was conceived, just in the space of a few minutes. I'd been looking out at the immaculate lawns (and enjoying some spectacularly tasty scrambled eggs on toast, and a really nice cup of tea), and mentioned to my dad that perhaps there could be some gaming mileage in using the idea of mowing as a basis (there was an entire class of games called "painter games" already predicated on the premise of traversing an area, "painting" it as you went, whilst beset by hazards). We tossed this idea back and forth between us for a while, grinning and adding various ideas for comedic hazards that could beset the player whilst mowing – an angry neighbour giving chase because he wants his mower back, an outraged gardener on the warpath if you ran over his precious flowers, a gormless dog (whose character was suggested at least in part by my mum's somewhat dim but sweet-natured Afghan hound) that would tend to chase the mower but who could be usefully directed against the pursuing neighbour)... in the space of a few minutes we pretty much had the design roughed out between us, even down to the theme music: staying in a quintessentially English place, we'd devised a very English game with a very English sense of humour, and so the accompanying

music simply had to be “English Country Garden”, didn't it? (And that tune was old enough to be out of copyright and so freely usable – so much the better).

The final bit of the design – the name – came to me as we were in the car on the way into the show. There had been advertising rivalry between Qualcast, who manufactured traditional, cylinder lawnmowers, and Flymo, who manufactured newfangled rotary hover mowers. One of the advertising taglines was “It's a lot less bovver with a hover” - and so our new game was to be called “Hover Bovver”.

Part 12: PESCLR

The design of Hover Bovver may have been pretty much complete in the space of those few minutes that morning near Birmingham, but before any coding could begin more immediate matters were at hand, most notably the show in Birmingham that was the reason for our being there. It began somewhat chaotically, with the realisation that we were one monitor short in our inventory (by now we'd acquired a reasonable collection on portable tellies and Commodore monitors to accompany the herd of Vics and 64s (and occasional Atari) that were deployed at every show. Me and my dad had to leg it off through town in the hour before the show opened, trying to find a computer shop or Rumbelows or somesuch that could furnish a new monitor in a hurry.

My brother Rob and his wife came up to visit us and to help out at the show (Llamasoft was entirely a family collaboration, and various brothers were likely to show up at the various shows to help out; although sometimes “helping out” meant locating the bar, providing a steady supply of beer to those in the front line on the Llamasoft stand, and getting plastered into the bargain).

During that show we were mainly promoting our Vic-20 catalogue, including Abductor, Gridrunner, Matrix, and Laser Zone. We had Gridrunner and Matrix running on the Commodore 64, but that machine was still relatively new and expensive at that time, and the Vic was still our main market. The C64 was gaining in popularity, however, and other programmers were getting interested in the machine.

By then we were also doing a small amount of work on the Atari 400/800 machines. These machines were certainly not mainstream in the UK, being rather expensive compared to the likes of the Vic and the Commodore; but, coming from Atari, they were quite enticing from a gaming point of view. They had excellent colour capability (being able to display colours from a vast palette of 128 colours, unlike most machines which could only choose from a paltry 8 or 16 if you were lucky). They also had some very interesting display modes allowing for a great deal of flexibility laying out the screen, and nice smooth scrolling like on the new Commodore 64. Being from Atari they also benefitted from a large library of officially-licensed arcade game conversions on cartridge, and they were well-supported by numerous third-party software houses in the US.

My initial Atari 400 had therefore been bought not with the particular intent of doing much software development on (although I

did do a simple little version of my Deflex game for it) – I got it purely as a games machine for me to play on. To be more specific, I got it as a Star Raiders machine.

I'd first heard about Star Raiders back in my University days, when I'd bought a copy of "Personal Computer World" whose cover featured the then-new Atari 400/800 machines. The title was "On Course For Atari" and it showed both machines flying out of what looked like a simulation of the display screen from the Starship Enterprise. Upon reading further I discovered that the starship viewscreen was in fact a display from the Atari game "Star Raiders", and the person doing the review couldn't find enough words with which to praise this particular piece of software.

Back then I was a poor University student and there was fat chance of being able to get hold of a 400 or 800 to play the game myself; I had to go for one entire term eating only baked beans just to be able to afford to buy a lowly VCS. However I never forgot about Star Raiders, and every now and again I'd see it referred to in the gaming press, and the verdict was always the same: Star Raiders was the best game, ever, on any system, in the whole universe, for ever and ever, amen. The normally sober BYTE magazine ran a special section in one issue devoted entirely to Star Raiders strategies and techniques. The game itself sounded like a blissful amalgamation of the old, relatively boring text-based Star Trek games (where you had to plod from sector to sector firing turn-based photon torpedoes at Klingon ships represented by asterisks and defend your home base ship) and my much-beloved Exidy Star Fire coin-op. There were still the Galactic maps and Klingons ("Zylons", in this game) and base stars to defend,

but everything was realtime and, astoundingly for its day, 3D and first person.

By the time Llamasoft was established and a bit of income started to come in, my mind turned again to Star Raiders and, being no longer an impoverished student, the bus was duly taken into Reading and an Atari 400 plus one Star Raiders cartridge duly procured. I must have read the manual for Star Raiders fifty times on the bus home; by the time I got there I already knew that F was for forward view and G was for galactic map and L for long-range scanners and H for hyperspace... I really couldn't wait to get home and finally play Star Raiders for myself.

Too much anticipation of something can be a bad thing. If you've had too long to dwell on the simple idea of how utterly great something is going to be, it can be a sobering experience to find, upon finally obtaining the item of desire, to find that in fact it's good but not as great as you'd hoped, and therefore feel ultimately a little let down and disappointed by it. And I'd been drooling over Star Raiders, on and off, for two or three years by then.

However, in the case of Star Raiders, the reality of actually owning the game lived up to every bit of my high hopes and expectations for it. I don't think that I have ever been so blissfully happy with any other single bit of software in my entire life. Star Raiders was, is, one of the best games ever made on any system, anywhere, ever, in the entire universe, for ever and ever, amen. For a couple of weeks after I got that Atari 400, my first feeling upon waking up in the morning was just one of absolute joy that I owned Star Raiders, and I could play it any time I wanted to.

Star Raiders was a little miracle; the size of the actual game code was a piffling 8K, but in that tiny amount of memory the designer managed to craft a game that was full of subtlety and strategy, not to mention fast, 3D space battles, excellent sound effects and even a good bit of humour (if you did badly you would be given some unflattering rating such as “Garbage Scow Captain Class 3” or “Galactic Cook Class 5”). Played in the ideal environment of a darkened room with the lights out and the sound turned up, the game was engrossingly immersive; you felt every impact of Zylon photon fire on your flickering shields, and actual panic upon having the shields fail under heavy fire and having to execute a dangerous emergency hyperwarp to get the hell out of Dodge and back to a friendly base star to get repairs done. The first couple of difficulty levels were fairly easy to master, but the only way to achieve the coveted rank of Star Commander Class 1 was to play the fiendishly difficult Commander Mission in which your shields were as fragile as eggshells and in which you had to carefully consider cruise speed and hyperwarp energy expenditure and make every shot count if you were to have a chance of succeeding.

To cram so much game, including then state-of-the-art 3D visuals along with fairly sophisticated scanner and status screens and a perfectly-balanced amount of strategy and resource management into such a tiny amount of memory must surely count as one of the best feats of game programming and design ever accomplished. The challenge was enough to keep you engaged for months, and to this day sometimes I will fire up an Atari emulator and play an occasional Star Raiders mission. It's worth remembering Star Raiders next time you hear Sony trying to convince us that good games cannot be made in less than 25 gigabytes of disk space...



“Star Raiders” for the 8-bit Atari home computers.

Anyway, I digress somewhat (but digression to extol the virtues of Star Raiders is always justifiable); I did own an Atari 400, but it was a spaceship rather than a development machine. However once Gridrunner became a success for HES, they asked me to produce an Atari version of the game (since the Atari computers were very popular in the US). At that point Llamasoft acquired one of the superior (but considerably more expensive) Atari 800 machines and a disk drive for development purposes, since developing software using the membrane keyboard of the Atari 400, and with a tape deck for storage, would have been ugly. I used this rig to bash out a version of Gridrunner for HES (unfortunately, and for similar reasons, just about as graceless as the C64 port had been) and subsequently, I must admit, to share in the disks full of illegal plunder occasionally to be found

when I visited the only other Atari user I knew in the whole Basingstoke area. Naughty, I know, but back at that time even if you wanted to buy Atari software it'd mean a trip to London at best or mail-order from the US at worst.

Although Atari was by no means a big market for us, we did have Gridrunner and my little version of Deflex for it that we could sell, and so the 800 came along with us to the Birmingham show as well as the herd of Commodores.

During the course of that show a couple of guys came to visit us on the Llamasoft stand – one of them had written a game and wanted us to have a look at it to see what we thought of it, and what we thought they might do with it. It was for the Atari machine, and I was curious, but not really expecting that much; I'd had various people show me works before and whereas many of them obviously showed enthusiasm, not that many of them were what I would consider to be of marketable quality.

However these two guys set about loading their game into my 800, and when it began and they gave me the joystick to have a go for myself, I admit I was completely blown away. Their game was excellent.

Basically the game was based on the Atari classic “Battle Zone”, in which the player sees a first-person view from out of a tank; he is free to drive in an endless 3-D environment (never quite managing to reach the volcano on the horizon) filled with geometric obstacles and enemy tanks out to get him. Battle Zone was a vector game, and due to

the 3D calculations and specialised display hardware necessary to recreate it, no decent home version had ever been made.

The two guys from Birmingham had managed to create a game that was, in many ways, even better than original Battle Zone. Rather than attempt to recreate the wireframe tanks of the original, they used simple solid geometric shapes that scaled smoothly to represent the enemy craft, and what appeared to be huge spheres to represent shots, both yours and those of the enemy. The pace of the game was way faster than Battle Zone and the action far more intense as you rushed hither and yon in your craft, blatting off shots at the enemies and dodging incoming shots of theirs. Shooting an enemy made it explode into a ring of clattering shards that spread out across the landscape. At the end of a level a portal would open up in the landscape and, upon entering it, you had to traverse a hyperspace full of fast-moving spherical objects that rushed by you at an alarming rate and which needed to be avoided; eventually you'd see a small doorway appear in the blackness of hyperspace through which a new world would be visible, and you'd drive out through it and into the next level. It was just excellent, and I told them so and let them demonstrate it on the Llamasoft stand. As for what they should do with it, I suggested that they should most definitely offer it to one of the big US software houses for distribution in America, and also sell it themselves in the UK if they could.

In the end they did a deal with Synapse Software, one of the most successful US software houses of the time, to sell the game in the US. And, after a bit of a delay due, if I remember correctly, to some wrangling over the rights to distribute in the UK, they released their game here, for the Commodore 64 as well as for the Atari, where it was

very well received as it deserved to be, and which helped to establish them as one of Britain's best-loved software houses.

The game was “Encounter”, and the guys were Novagen.

Bless 'em.

To Be Continued

The History of Llamasoft is an ongoing project; this is the story so far.

Check the Llamasoft website at

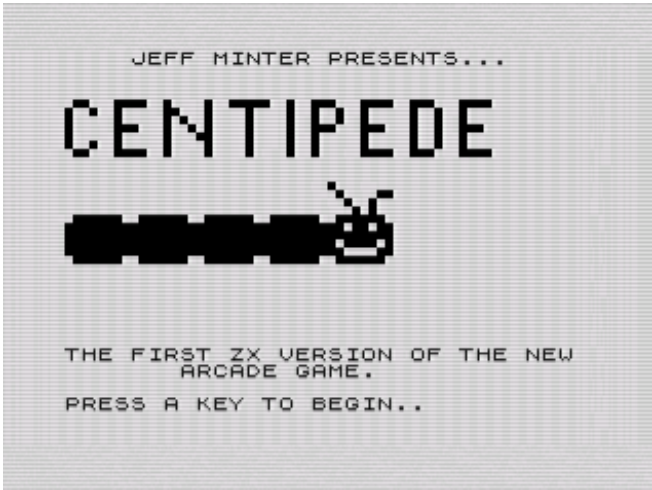
<http://www.llamasoft.co.uk/>

for more chapters as they are released.

A History Of.... Gridrunner

Gridrunner was one of Llamasoft's earliest games, appearing in 1982 for the Vic-20 and Commodore 64. The title has been revisited and reimagined at various points over the years, and the release of Gridrunner Revolution marks the endpoint of a development arc that has spanned no less than 27 years.

In fact the history goes back a little further than that. It's plain that in its earliest forms Gridrunner was at least partly inspired by the Centipede coin-op arcade game, and back in 1981, before Llamasoft was founded, I created a Centipede game for the Sinclair ZX81.

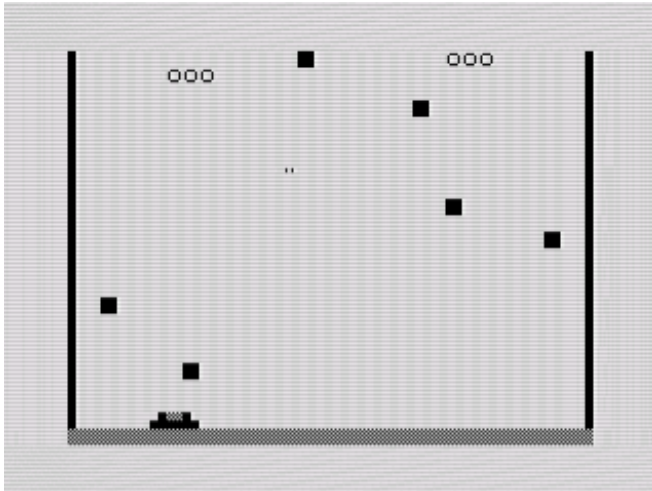


“The first ZX version of the new arcade game”? My arse. I'd never even played the arcade game at this point.



Haha, “JCM.SOFTWARE” - was there ever a bedroom programmer who *didn't* do such stuff?

I say that I created a Centipede game rather than that I did a version of Centipede, because as a copy of the arcade game the ZX81 version would be pretty poor. At the time although I'd heard of Centipede and even seen pictures of the screen I'd never actually played it, so my version was rather inaccurate. It had "centipedes" and blocks that caused them to split and turn like the mushrooms of the coin-op, but the player ship was a Space Invaders-style laser base that only moved left and right along the bottom of the screen, and the Centipedes dropped bombs.



Actually I think they dropped full stops on you. And you fired quotes at them. CUTTING EDGE GRAPHICS BABY.

This deviation from the original inspiration continued with the design of the original Gridrunner. I was working on the Vic-20 at the time, and Centipede seemed to be the game of which everyone and his

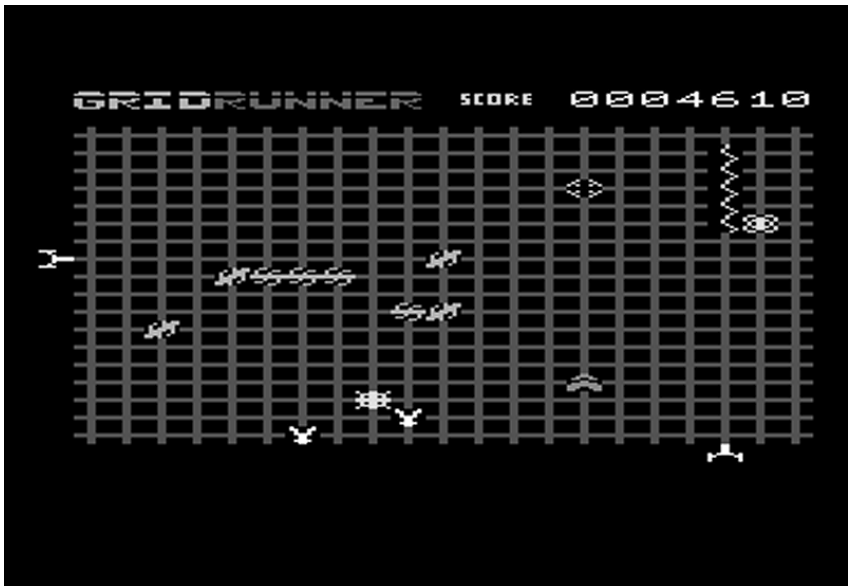
dog decided to do a version on that platform. It was also the time when Atari were waking up to the fact that many small developers were making free with their IP all over the various home computing platforms. Not surprisingly, Atari started to make noises to the effect that they would deal litigiously with anyone producing “clones” of games to which they had copyright.

Although I still fancied making a game in the style of Centipede, these two factors led me to come up with a design that was significantly different from Atari's bug-themed coin-op. Sufficiently different, I hoped, to avoid Atari's legal wrath, and also to mark it out from the many Centipede versions that were already appearing for the Vic-20.

The name and basic theme came to me whilst travelling on the London Underground one evening after attending a computer show in the city. Llamasoft used to exhibit at many such shows during the course of a year, and we'd stay in a hotel near the venue for the duration. Most evenings we'd take the Tube to the West End and spend time hanging out and playing games in the many arcades that were to be found in the Soho area.

At the time the excellent film “Blade Runner” had just come out and, as I was standing on the platform waiting for the Tube, I glanced at a large poster advertising the film. My eye came to rest on the title BLADE RUNNER in that distinctive red font, and for some reason the name GRID RUNNER came into my head. No idea why, I wasn't actively thinking of grids or games or anything at that point; maybe it just sounded good to me.

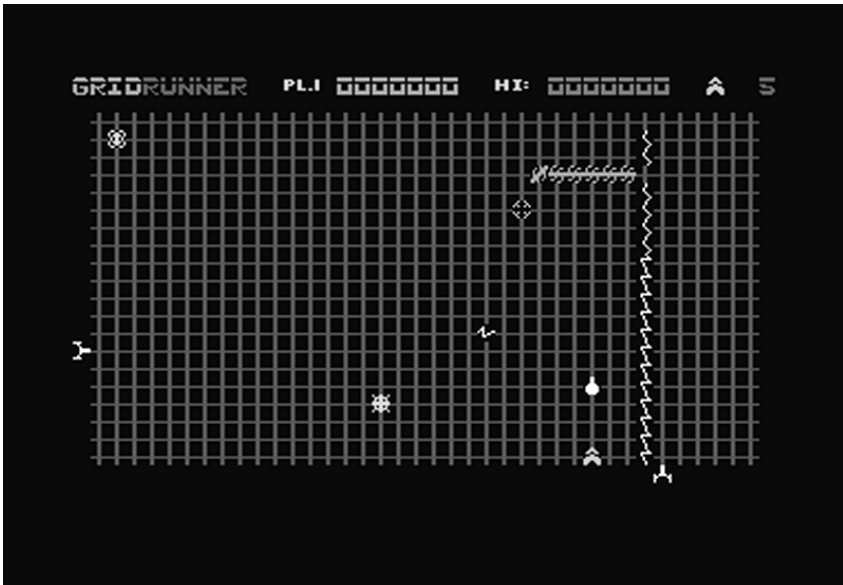
This proved fortuitous for my plan to produce a less Centipede-like game, since it did start me actively thinking about grids and games. Instead of bugs in a garden the action would take place on a grid (maybe the “Game Grid” from Tron was in the back of my mind). Instead of centipedes there would be snake-like “droids”, and instead of harmless mushrooms there would be “pods” which, if not dealt with, would turn into bombs which would fall down the grid towards the player.



Gridrunner on the Vic 20. Fat CAN be sexy.

To further distinguish the game's design from Centipede, and in keeping with the new game's more hard-edged, sci-fi-themed design, I added lasers. Any game is better with lasers, I figured. Two laser

guns traversed the grid's edges and every few seconds they would fire - one creating a large vertical beam from top to bottom of the grid, and the other firing a small pulse horizontally across. Where the two beams intersected a new pod was formed, and if the player was caught in the beam or hit by the pulse he would lose a life.



Gridrunner on the Commodore 64. WAY too skinny, I like a bit of meat on my pixels.

The rhythmic pulsing of the lasers and the deep electronic throb that sounded with their firing added a nice element of tension to the game, and provided it with a distinctive “heartbeat” during play. These lasers were to be one of the defining constants, albeit in a variety

of different forms, that defined the various Gridrunner games as the design evolved over the years.

The Vic-20 game proved to be quite popular, and in very short order the game was ported to the Atari 8-bit and the new Commodore 64. Of those early versions I think the Vic-20 version is my favourite. The game's graphics look a bit small and weedy on the higher-resolution Commodore 64, and the Atari version was a rather rushed port with graphics that are too chunky and rather ugly. The game was designed primarily for the Vic and although the graphics are large and low-rez, to my mind it is the Vic version that looks and plays the best.

(You can play and judge for yourself - recreations of both the Vic-20 and Commodore 64 games are available as unlockable extras inside Gridrunner Revolution. Just play ten levels in Easy difficulty to get the Vic version).

The next game in the series wasn't long in coming - just a few months, in fact. The original Gridrunner was created for the unexpanded Vic-20, which had only 3.5K of available memory. This made for a rather sparse and somewhat repetitive game, with successive levels being identical to the previous ones, just a bit faster or with more enemies and faster lasers. The basic design worked well, so I decided to make an extended version for Vic users who had a memory expansion unit.

In this extended version I was able to add a lot more variety as the player progressed through the levels. In addition to the regular boustrophedon progression of the enemies as in the original Gridrunner (and Centipede), they now could attack and race across the grid in

diagonal formations. The whole grid was animated through simple character redefinition, and various different grid shapes appeared on different levels. Some levels had no grid at all.



Camels join the Droids in the 8k Vic-20 version of “Matrix”.

Another Llamasoft obsession - camels - made their first appearance as enemies in the game, and on some levels deflectors (taken from an earlier Llamasoft game, “Deflex”) could divert the player's shots. It was possible to earn special bonuses for killing an enemy with a trick shot off the deflectors, for sparing the life of a camel, and for a variety of other moves that the player had to discover “almost like an early version of Achievements.

The regular pulsing lasers were retained, and on some levels a new (and annoyingly unshootable) enemy called the Snitch would

appear. The Snitch walked along the top edge of the grid, and if the player remained in one place for too long the Snitch would “tell” the lasers to fire out of sequence and shoot immediately if they were passing the player, killing him instantly. Players regularly cursed me for putting the Snitch in the game.

All these additions made for a much more enjoyable and engaging play experience, with much more to shoot and do, and less repetitiveness between levels. Once again the game was ported to the Commodore 64, and once again it was a bit of a rushed port, and just as with the original Gridrunner I think the Vic-20 version looks and plays the best.



A gridless level with deflectors in Commodore 64 “Matrix”.

This game was sold under the name “Matrix” in the UK and Europe (and I am still waiting for my royalties from the films). And, somewhat confusingly, as “Attack of the Mutant Camels” in the US. Confusingly, because Llamasoft released another, completely different game on the Commodore 64 and 8-bit Atari called “Attack of the Mutant Camels” - the story about the reason for that is told in the *History of Llamasoft*.

There was one last variant of Gridrunner - by now quite well distanced from its Centipede origins and an entity in its own right - before the end of the 8-bit era. I was approached by Mastertronic to create a budget title for them, originally for the Commodore 16 (a somewhat strange machine introduced for people who for some mad reason might want to buy something that wasn't a Commodore 64, and which wasn't in any way nearly as good).

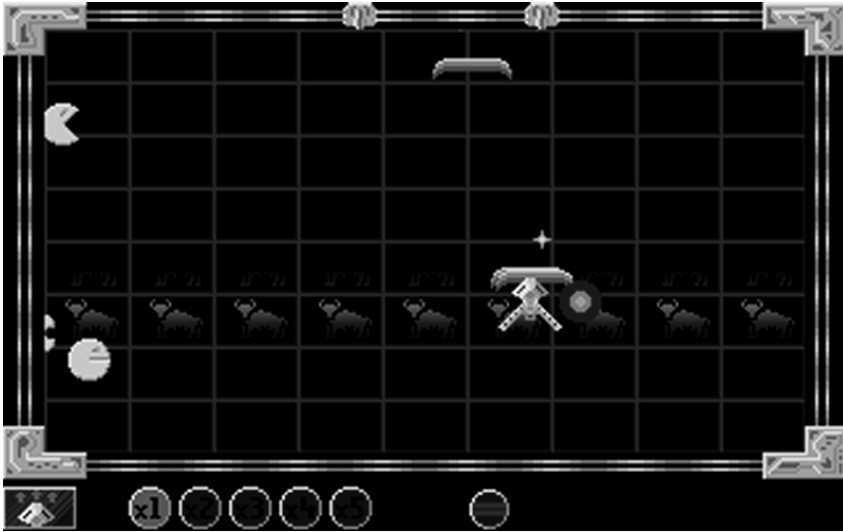
I made a game that was similar in parts to Matrix, with greater variation in how the chains of enemies could move and where they could appear from. Pods and camels were there, and the travelling laser, in this case a bar that moved steadily down the screen, occasionally pulsing. The most significant change was that the player now controlled four rapid-firing ships at once, moving them as a squadron in a variety of formations. Attacks came from all directions, befitting the fact that the player ships now had a lot more firepower in many different directions.

This game was released as “Voidrunner” by Mastertronic. I made the customary port to the Commodore 64, and Mastertronic made ports for the Spectrum and MSX. Whereas the game was certainly a

nice enough budget release I don't feel that it was quite as good as Matrix on the Vic-20.

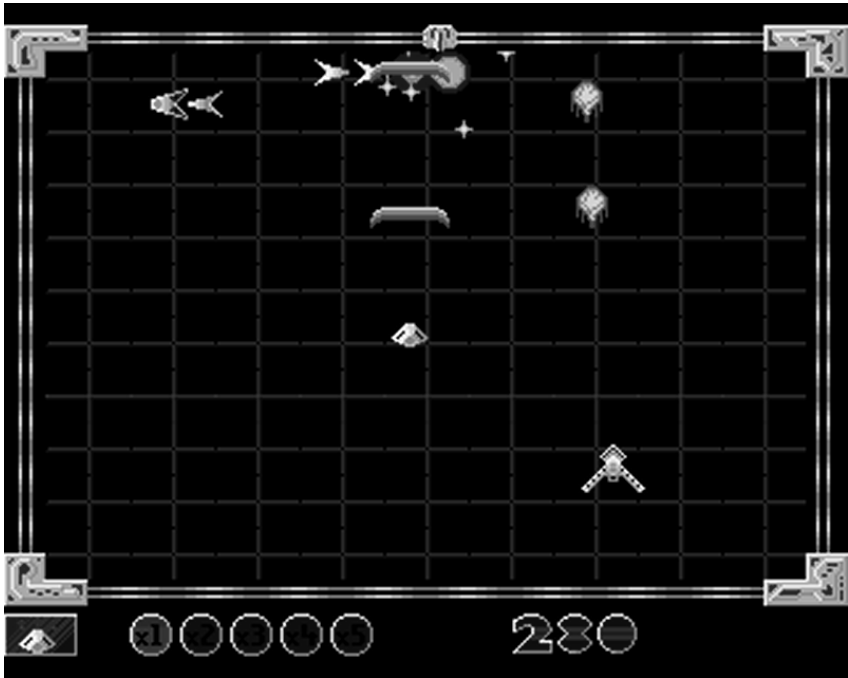
The next visit to the Grid came in 1988. I'd spent the start of the 16-bit era concentrating on developing my light synthesizer ideas on the Atari ST, and having made "Colourspace" and "Trip-a-Tron" I wanted to make a game on the ST. I spent a while getting together some sprite, tile and text routines and then decided that, as Gridrunner had been fairly well known during 8-bit times, it'd make sense to do a new version for the new era.

With the luxury of 512K to spread out in, and much higher resolution graphics to draw my sprites with, I could create a much more extensive game than the original Gridrunner and Matrix. In truth, higher resolution graphics were at that time a mixed blessing for a lone coder/designer like me. On systems capable of only low-resolution, chunky graphics there's only so much you can do, so it's possible to get away with fairly primitive shapes and, due to the fact that nobody else's primitive shapes look much better, nobody notices that you're actually pretty rubbish at drawing artwork. On the 16-bit machines the graphics had improved to the extent that good artists could indeed produce excellent work and much "coder art" such as my own looked a bit shabby by comparison. Not that it was game-spoilingly terrible; just that it looked rubbish compared to "proper" art and it was obvious that I was a coder bashing pixels together in Neochrome rather than someone who could, you know, actually draw.



Gridrunner on the Atari ST. Make things look metallic to try to cover up how crap your sprites are.

Nonetheless I had a good time making the game, which turned out to be like some kind of mashup between Gridrunner and Revenge of the Mutant Camels, with the player beset not only by squadrons of arrow-shaped droid ships, but also various other “wacky” enemies (I distinctly remember that one level featured a rain of skis). Shot snakes would sometimes spontaneously explode, with each segment changing into a collectible bonus item - in some cases fruit, and in some cases tiny llamas that one had to be careful not to shoot. (Collectible beasties would soon be making an appearance in another Llamasoft ST game - “Llamatron” - a couple of years later).



Screen from the Amiga version of Gridrunner showing the detachable nose cone of the ship.

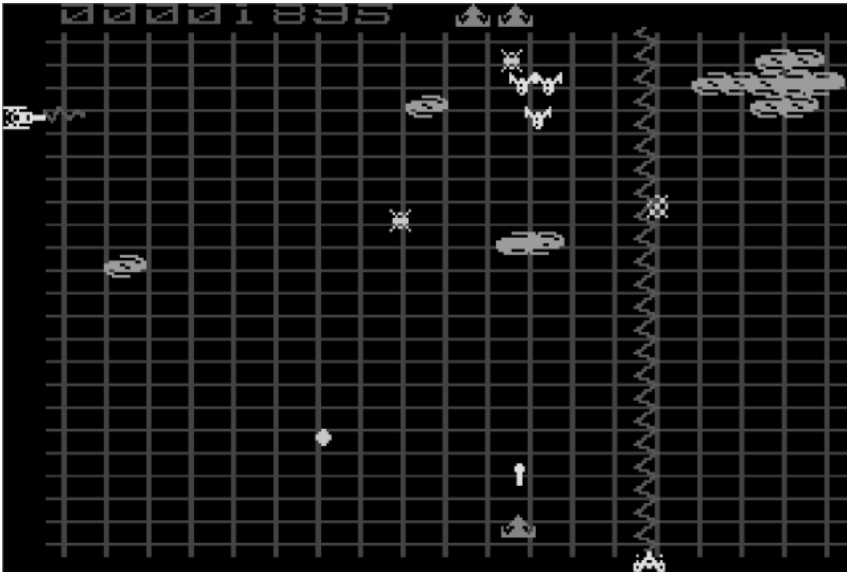
The scoring method borrowed from pinball in that it featured a “Bonus Multiplier” which started out at x1 and which would steadily increase through good play up to x5. Losing a life caused it to reset to x1. I liked that method in pinball, where it made hanging on to your ball for as long as possible an even more urgent imperative, and I liked how it transferred to video game play, reinforcing the basic underlying videogame imperative *to stay alive, no matter what*.

The single biggest design change was to the player ship. For one, it was now mouse-controlled, allowing the player to nip with alacrity to all areas of the playfield, far more swiftly and precisely than was possible with a joystick. The ship was in two parts - it had a

detachable nose cone which you could leave somewhere on the screen, firing independently, while you busied yourself elsewhere, and then recall it to you by holding down a mouse button when you needed more local firepower. You could even place it ahead of you up the screen and then fire into it from below, causing augmented shots to emerge.

The lasers at the sides were there, many of them this time, travelling on little tracks at the screen edges and firing pulses rather than beams. The game design overall had been expanded and extended to a scale commensurate with the more capable platform the ST provided, but it still contained elements to tie it to the original Vic-20 game and define it as Gridrunner. I also ported the game somewhat roughly to the Amiga - it worked fine on that machine but didn't really use any of the Amiga's special features, it was really just a case of making the ST code work on the Amiga and having done with it.

There was one more version of Gridrunner on the Atari ST which I shall mention for completeness - in truth I haven't seen it for years and I'm not sure if it's in the archive anywhere. I can't remember why I was asked - for a coverdisk or for a demo I guess - but I was asked to do a game in 4K. Since the dear old unexpanded Vic had been 3.5K I thought it would be natural to choose Gridrunner as the subject of the demo and so I made a little version that fit in 4K on the ST. It was super primitive but it was just about fully functional.



Found by a Yakyak forum member - “Gridrunner” in 3.5k on the Atari ST.

There was almost another version hidden in “Defender 2000” for the Atari Jaguar. I was living in the US at the time, and one 4th of July I decided that as I was a Brit I couldn't possibly take a day off with the colonials as they celebrated kicking our Great British arses, so I'd work all day instead and do a “4th of July” mix of Gridrunner to hide away in D2K. I got it about half done but then it wasn't really turning out the way I wanted it, and I didn't have time to work on it more, so I never finished it.

Gridrunner arrived in the 21st century in 2002. At that time I was developing a few small games for the Pocket PC platform - it was

actually pretty enjoyable to be doing some short, simple, fun games that could be coded in two or three months as opposed to major development work which can take years these days. I'd revisited a couple of old Llamasoft titles already, updating "Hover Bovver" and "Deflex" for the handheld device. I was using a wrapper library at the time which effectively enabled you to produce not only code for the target device but also a version that would run on a kind of simulation of the PocketPC environment, in a window on the PC. This was a bit rough and ready and intended more as a development aid to allow you to test things locally on your PC without having to keep downloading your code to the PocketPC rather than as something to develop proper PC titles on. These PC versions were quite playable though, and since they came almost for free it seemed silly not to use them and offer them for sale alongside the intended PocketPC versions.

I started developing a version of Gridrunner in the same framework, intending to produce two versions, one each for PocketPC and PC as before. However as the design took shape it became clear that I'd made it too hard for the PocketPC to manage well, so I finished it off for the PC only, albeit within the context of this rather strange PPC-simulating wrapper.

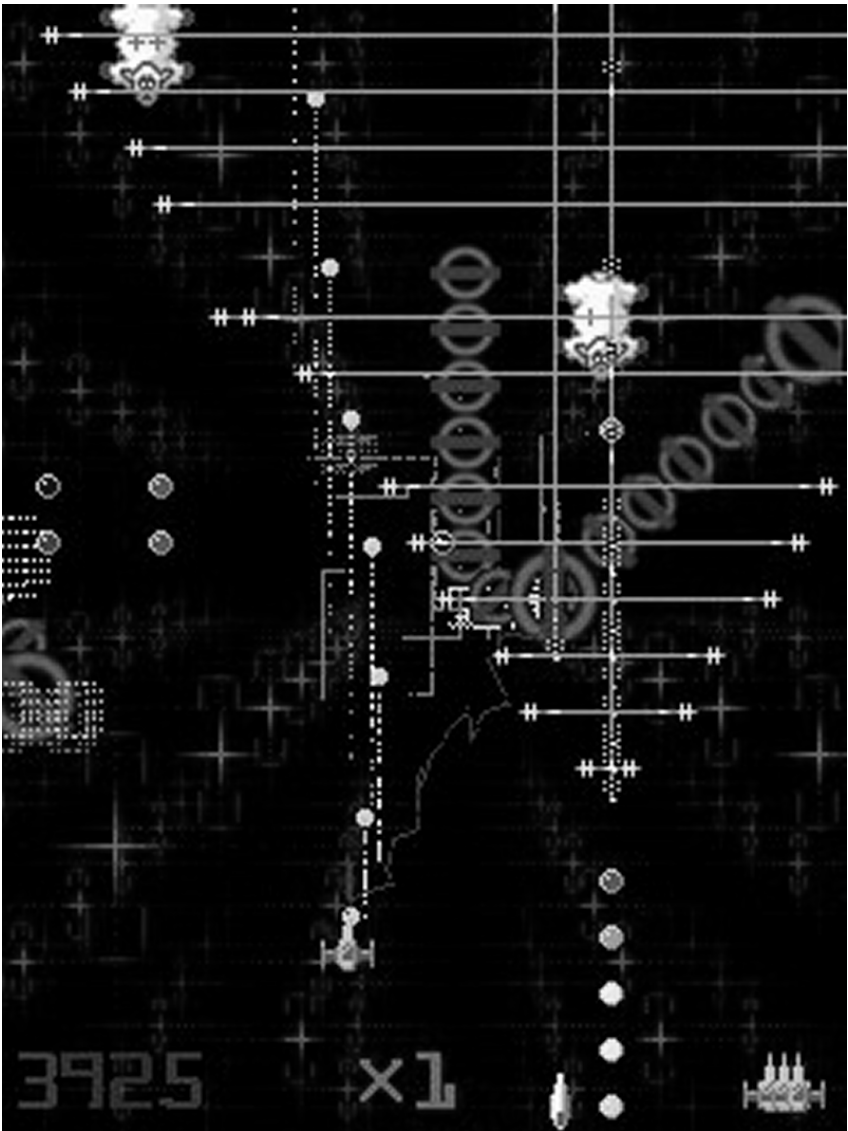
The Yakyak forums were just opened at that time and I shared the early development of the game with the users, offering up early stage works in progress so that people could see what I was doing. At first I almost wished I hadn't because the early versions actually weren't that great to play with. Some ideas were there but nothing was balanced, nothing felt right, nothing looked how I wanted - there's an awful lot of tweaking and tuning that has to happen sometimes before you know for sure that something's on the right track. It's not a good

idea to show stuff or offer it up for people to see until after that point is reached, and at first people were disappointed that it didn't seem to be turning out that great.

There is often a tipping point where you make a few adjustments over a few days and suddenly the parts begin to mesh and you get a glimpse and a feel of what it was you were looking for in the first place. For me that moment is when a game gets its soul and comes to life. As a developer it's also a great relief to reach that point, for it is then and only then that you know that everything is going to fit together and work well, and from that point on you know much more clearly which direction you're working in.

There was a particular three day period where that transition happened back then, and I created a demo with just a few levels and hardly any graphics data yet defined and offered it up for people to play with. Everyone else could feel the change too and people could feel that the game was going to turn out to be nice.

The forum also helped me circumvent the problem of “programmer art”, since there were people in the community far better at drawing sprites than me, and I encouraged people to contribute sprites for the game. (Curiously enough this was actually the second time I've had sprites contributed to a game by an online community. It originally happened in the mid-80s, on Compunet, an old Commodore 64-based online community. A bunch of us became regulars in the chat space of an evening, chatting and swapping screens of chunky Commodore graphics and even little quickly hacked up games and demos. I asked for sprites from the regulars there, and they ended up in my Commodore 64 game “Batalyx”).



Gridrunner++. In which the player rescues sheep whilst beset by lasers and gangs of Underground signs. WACKY.

It was fun working with the community in that way, and as well as contributing sprites there were many willing hands to help with testing and debugging the game prior to release. The game was released as “Gridrunner++” and was surprisingly well received in the press for such a little game.

The game itself - although once again expanded from its previous incarnations - still retains some elements of the ur-Gridrunner; the snakelike enemies, the pods that turn dangerous if not dealt with straight away, the regularly pulsing lasers (now adrift from the sides of the grid, on a free-floating cube that periodically unleashes crossed bolts across the playfield). A rescue mission is added to the gameplay in the form of fluffy sheepies that periodically float down the screen and which must be saved.

As each sheep is saved the power of the ship's weaponry increases until finally “the Pill” emerges, a giant sheep's head that smashes into enemies and which can itself be powered up to emit lasers and rush frantically around the screen killing large amounts of baddies.

Much though I like that version - and to this day it stands up as a challenging and satisfying little shooter - I always felt that, running inside that weird little wrapper framework and not really using any of the power of modern graphics hardware, it wasn't really posh enough to be considered as the true, native PC update that I felt Gridrunner deserved. I felt that one day it would be nice to develop Gridrunner into a proper, full-scale PC game and bring it thoroughly up to date.

And thus we arrive, after a long journey, at Gridrunner Revolution.

Why that name?

Well, for ages while it was being developed it was referred to as “Gridrunner+++”, which was a handy way of referring to it but a bit silly with all the pluses. It's also a significantly different and new game, so it deserved its own nomenclature rather than just a one char concatenation onto the name of an existing game.

The name is also a bit of a smirk about the time when the Wii was still referred to as the Revolution and there was talk of games called [something] Revolution all the time - but mostly the name refers to how it's played. Like the Atari ST and earlier PC version of the game, the mouse controls the player's ship; the mouse feels fluid and natural to use to move the ship in this kind of game. However now the player can also rotate his ship to face danger or to engender rewarding bullet trajectories - hence “Revolution”.

And yes, some of the threads that tie all the versions together across the years, and which lead back to that dear old Vic-20, are still there - there are snake-enemies and boustrophedon motion. There are Flowers instead of Pods, and they can still get dangerous if you're not careful. There's a free-floating Sun that regularly fires and emits lasers, giving a cadence and heartbeat to the game just as the traversing laser guns did on the Vic in 1982. And sometimes you can see that underneath all the fireworks and glory - the grid is still there.

So what's new in the design? Well - that I'll let you find out for yourselves.

Gridrunner Revolution is both ancient and modern, bringing gameplay in the style of the old school, and with 27 years of development along the way, into futuristic focus on today's graphics hardware. Procedural generation of enemies and backgrounds in the latest and most advanced incarnation of the outstanding Neon graphics synthesizer means that there's no more worries about “programmer art” - Revolution looks astounding and sounds awesome too with plenty of excellent tunes to accompany the exhilarating and immersive action.

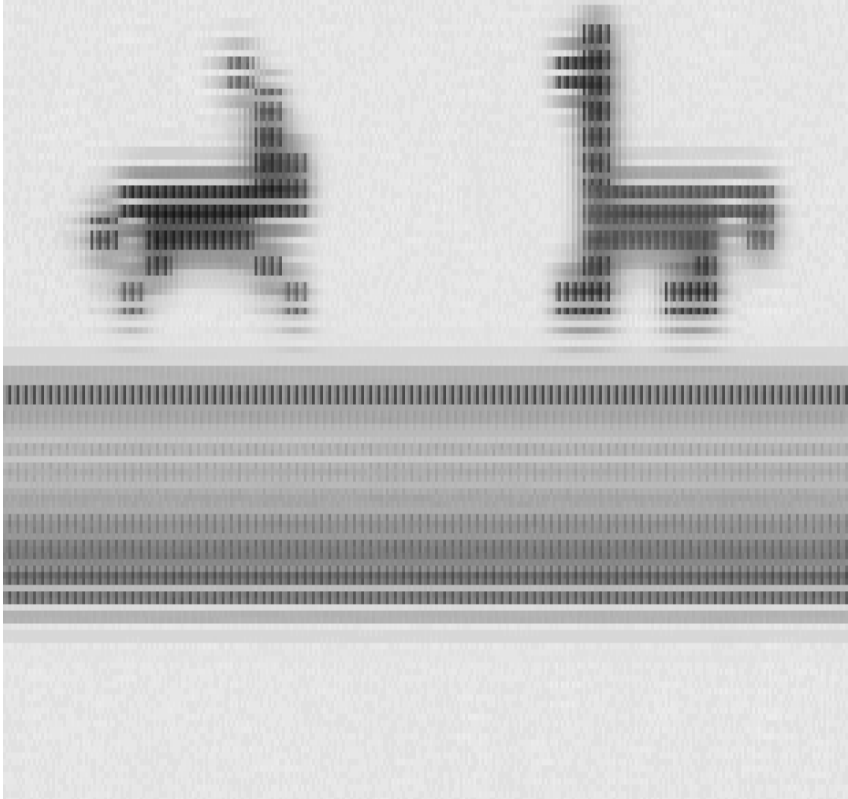
I will write more about the design of Revolution itself in due course.



REVOLUTION

visual synthesis

the evolution of
llamasoft's graphical style

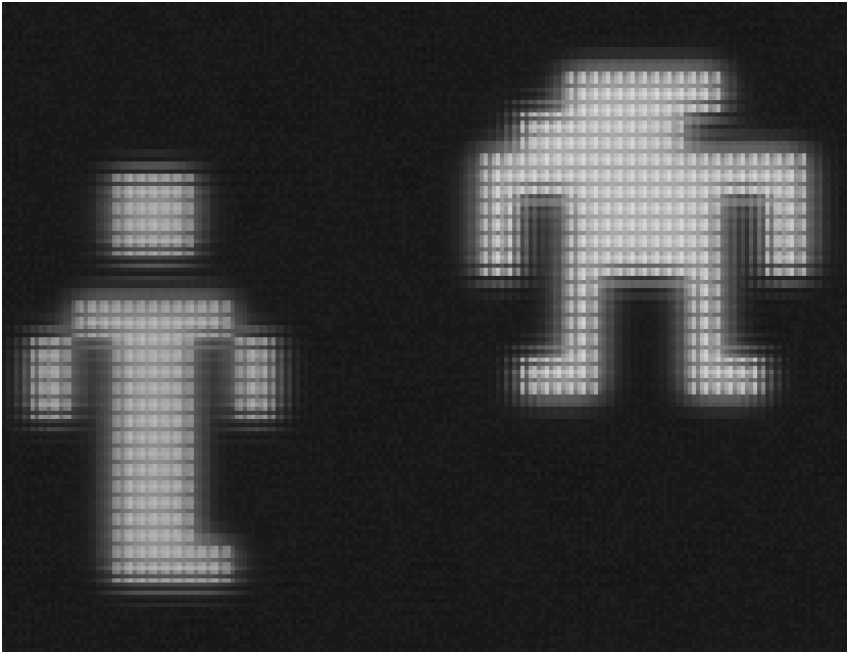


Back in the dawn of time the very first videogames were perforce almost entirely abstract. The designers of those first primitive systems couldn't do much more than generate a handful of white lines and squares on the TV screen, and gamers had to use a hefty dose of imagination to interpret these primitive visuals as being the various sports games after which they were often named.

Towards the end of the 1970s, many coin-op videogames and, perhaps most significantly, the Atari VCS emerged. These systems were unlike their predecessors, which typically generated the game elements by using hardwired circuitry. These newer systems used a microprocessor and ROM memory to store programs and game graphics, allowing for much more diverse and interesting gameplay and for far better graphical displays than their hardwired ancestors.

Looking back from a 21st century perspective these second-generation game systems look laughably primitive. We take for granted the generation of near photorealistic gameworlds in high resolution in realtime. And yet there is a certain nostalgia for the old pixel graphics from back in the day.

There was a certain distinctive, futuristic aesthetic that emerged from those primitive display systems - a feeling, an idea that was in part a product of the very limitations of the machines generating the displays and the very screens upon which the displays were projected. Everything was formed of glowing geometric primitives - be they the component pixels of raster displays or the neon purity of the arcade's vector displays. Colours were bright and vibrant. Objects seemed formed of pure light.

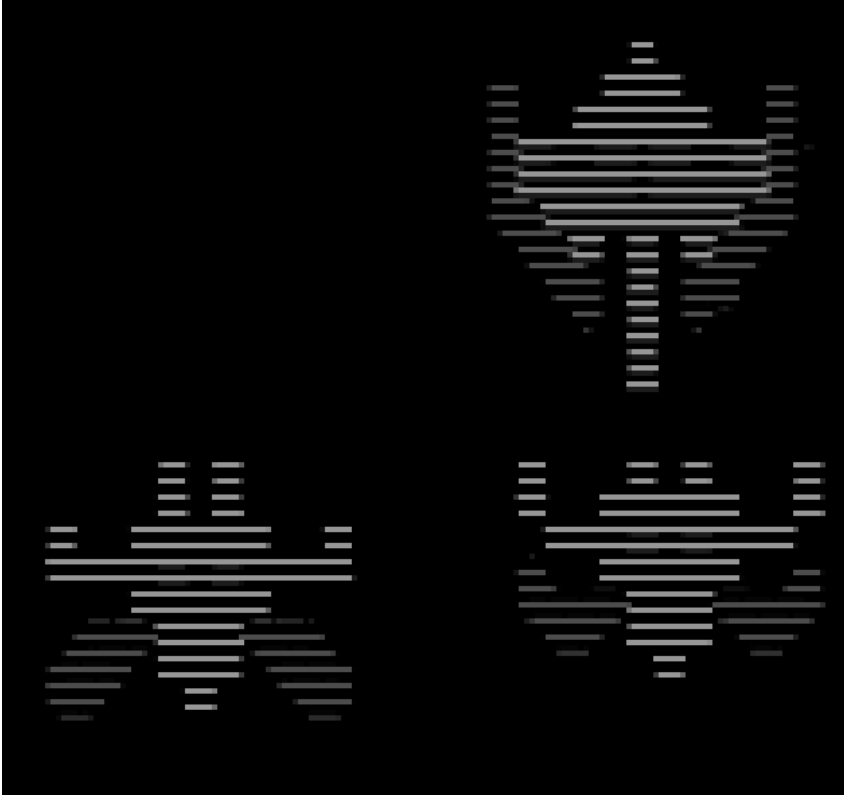


Look closer and below those fundamental, atomic structures there were visible finer atoms still, in the form of the individual phosphor dots and shadow mask apertures of the CRTs upon which these games generated their displays.

From our modern perspective we know how displays and graphics evolved, and what we have now is largely "realism". Part of the work of modern graphics systems is precisely to spoil and perturb the geometric purity of generated graphics - realism is the goal, and reality isn't often geometrically perfect. The eye needs to see roughness and nonuniformity if something is to appear real.

Back then, though, when we thought about the future of computer graphics, we didn't end up thinking of anything as mundane as simple reality. We projected what we had forwards and ended up

visualising strange future virtual worlds composed of neon-glowing primitives - the world of "Tron" and its ilk, and the inspiration for Gibson's cyberspace. The reality of living in the future is actually rather mundane. Cyberspace is just a bunch of web pages. Cutting-edge graphics just look like the world outside.



For me there was a lot of charm in the oldschool, primitive graphics. They were interesting in the same way that the sounds of the first electronic music synthesisers were interesting - in part because of their primitive geometric construction. There is little precision in Nature, so the first time one heard square waves and precise filter sweeps, for all that they were primitive, the experience was new and

intriguing. The same was true of graphics, back in the bright dawn of the 8-bit day.

Like graphics technology, the technology of music synthesisers has evolved tremendously since the days of Pink Floyd and Gary Numan. Modern music synthesisers are capable of emulating the sound of any traditional instrument as a matter of course. But they are not used only to do so. Musicians also use them to generate sounds that could only come from programmable electronics. Many people find these entirely synthetic sounds just as interesting as those of traditional instruments, if not more so; and musicians use them to create a much wider variety of types and styles of music than has ever been possible before.

Modern graphics technology, however, is almost always used to generate plausible conventional realities. We may use it to extrapolate and depict possible futures, but in general even when we do so they tend to be possible futures of our own conventional reality. There may be spaceships, aliens and laser battles, but by and large CG reality still looks like reality.

This has inevitably had an influence on the nature of game design since the early days. Games strive to be more and more "real". People expect believable worlds in which to move about, they expect to interact with realistic characters and, unless taking part in some sports simulation, to be engaged by some kind of a story. This direction of game development has led to some truly stunning experiences becoming available to players - the term "interactive movie" which once seemed so futuristic and awesome now seems inadequate to

describe experiences such as those provided by large-scale MMORPGs. "Interactive world" seems closer to the mark.

It is without question that this direction of game development has yielded some fantastically rich and engaging play environments, but it also has its costs. Creating such deep and detailed worlds takes a lot of resources. Modern games have production budgets on a similar scale to Hollywood movies and require large teams working for several years to make them.

Largely gone from modern mainstream games is the idea of a single creator making a distinctive mark upon his games. There are still well-known designers (the likes of Shigeru Miyamoto and Eugene Jarvis spring to mind) but the more distant they are from the work themselves, and the larger the teams they work with, the less recognisably they are able to give a distinctive style to a game. Much of modern gaming looks homogeneous: it's very difficult these days to look at a game and to be able immediately to identify which team or company made it.

There are times when players don't necessarily want to sit down and engage in a deep, story-led gaming experience for an hour or two. Perhaps some of the reason for there being a degree of nostalgia for "retrogaming" is that sometimes you don't want to engage deeply with complex worlds and characters. Sometimes you're just looking for a little twitch-gaming action like the old arcade and console games used to supply.

So modern gaming is awesome, it's true, but in some respects it's limited. Development of mainstream games is expensive and

requires a lot of resources and manpower, putting it out of the reach of individuals and smaller teams. Large teams are expensive so developers are more inclined to play it safe and take fewer risks in game designs, which leads to many similar-looking games proliferating in just a few genres which are known to be successful. Within this milieu individual creativity is nigh-on impossible and the kind of playful creativity which used to be highly prized as part of a game's design seems less relevant.

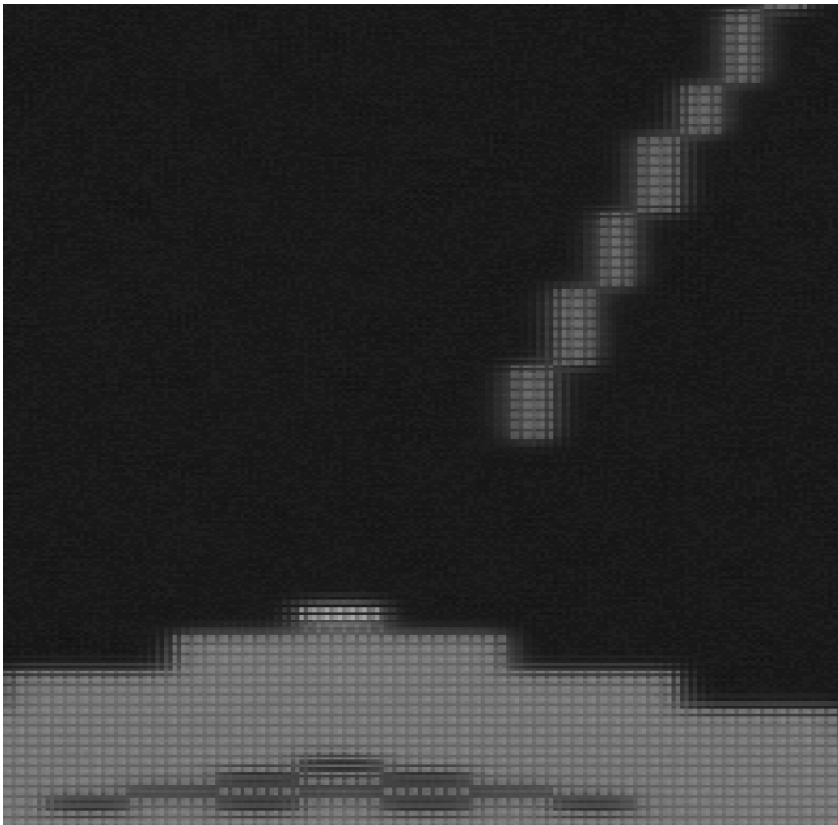
This state of affairs has arisen at least in part because of the direction that the evolution of graphics hardware has taken gaming - relentlessly towards greater and greater realism, along with the concomitant complexity of making believable realistic virtual worlds.

As a small, independent game designer I don't want to be constrained by realism. When I start designing a game I don't want to have to first of all find artists to design the realistic stuff I'm incapable of drawing myself. I don't want the chore of having to create characters and populate simulations of conventional reality. Often I am not even that interested in telling a story.

I am much more interested in bringing up to date the kind of gaming experiences we used to have back in the days of glowing geometric primitives and abstract neon gameworlds. A good arcade game from that era was a simple but beautifully constructed thing. Often a few simple rules were sufficient to define the game, but despite their simplicity the way in which they fit together and interacted with each other and the player's control inputs yielded a rich and satisfying

complexity which good players could understand and use to refine their techniques, allowing for longer games and better high scores.

Once you mastered a game you had a feeling of truly being "at one" with the game - understanding its every nuance, able to respond immediately and effectively to situations as they arise during gameplay, able to play for more than mere survival - able to play beautifully rather than simply not dying. Being in that state of oneness with the game - a state players refer to as being "in the zone" - is almost like meditation. You feel connected to the game, and to the mind of its designer, in a way that is personal and profound and almost entirely lacking in modern, large-scale, realistic story-telling mainstream gaming.



For this kind of immersive, meditative gaming experience the enforcedly abstract nature of the onscreen graphics seemed entirely appropriate - leading to a feeling of taking part in an activity in a space entirely different and separate from conventional reality.

However when we turn to retrogaming looking to rediscover this experience, in all but a few cases it is often less than satisfying. Out of their time, and displayed on modern hardware, the old graphics lose something of their power - they are simply too primitive for the old magic to work on eyes jaded by today's super high definition displays.

So - is that kind of experience lost forever? Not necessarily.

Let us return to the idea I touched on briefly earlier - that in their geometric simplicity, early graphics displays were somewhat analogous to the sounds made by early music synthesisers. In music synthesis the precise, electronic, procedural aspect of music generation was developed further as well as the ability to emulate conventional musical instruments. Various genres of electronica took their place alongside more conventional methods of music making. The gamut of musical creativity is much broader than before as a result, and encompasses everything from orchestral compositions performed by many people playing many conventional instruments through to individuals using programmable hardware to create "instruments" and music entirely distinct and of themselves. When you hear Aphex Twin, you know who the composer is just because his individual style and usage of the technology is so distinctive.

The drive of modern graphics hardware almost exclusively towards realism has left us with a situation where almost all of modern gaming is of a specific type, in a particular kind of setting, and "performed" by the gaming equivalent of an orchestra - a large development team.

But what if we started to use graphics hardware in a different way? What might happen if we used the enormous computational power of modern graphics hardware as a means to build on those early geometric primitives and to develop them further, to revel in their procedural nature and mathematical precision, to extrapolate them in directions completely orthogonal to reality?

I believe that in doing so we can create the gaming equivalent of electronica to counterpoint the existing "orchestral" style of mainstream gaming. It also allows for smaller teams or individuals to partake in the process, since fewer resources are necessary to create abstract synthetic graphics; and for them to create their own distinctive style, since the creative possibilities of procedural synthesis are pretty much infinite.

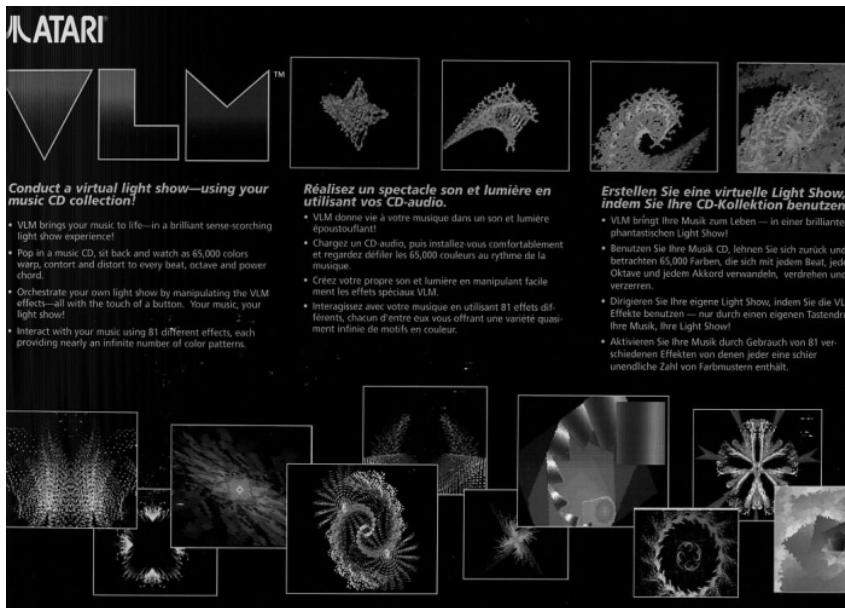
There are already examples of this approach which have been notably successful - take for example the likes of "Rez" and "Geometry Wars". Both these games take as a starting point the "feeling" of 80s graphical futurism - the world of "Tron", the sparse neon beauty of vector graphics - and use graphics hardware to bring these ideas to an updated reality. Coupled with the underlying mechanic of a decent oldschool arcade-style game, both Rez and Geometry Wars proved that this style of gaming can have a place in the modern world.



I took my own first steps in this direction in 1994 with "Tempest 2000" on the Atari Jaguar. I already had a fairly long history of experimenting with the creation of synthetic abstract graphics, due to a long standing interest in creating an interactive "graphics synthesiser" to accompany music, a predecessor to the kind of music visualisations that are now commonplace.

I began with some simple experiments on the Commodore 64 back in 1984, and by the beginning of the nineties I was working with a couple of other guys on a Transputer-based interactive music visualisation system that we called the Virtual Light Machine, or VLM.

When I came to do "Tempest 2000" it seemed natural to use the same kind of ideas for in-game effects as I was using in the VLM - the original coin-op "Tempest" was already beautifully abstract and there was no need to approximate any form of realism in taking the design forward.



The game when completed was very well received and is considered to this day to be one of the defining titles on that platform. The graphical techniques I used for the game were further developed into the music visualisation system built into the Jaguar CD-ROM unit, which I also called VLM. The techniques developed in Jaguar VLM inspired much of the first wave of music visualisers which started to emerge later in the 90s on the PC and Macintosh.

Since then this style of game development has been very much my main interest. In 2000 we released "Tempest 3000" and VLM-2 on the NUON enhanced DVD system, but these are unfortunately little known due to the unfortunate lack of uptake of the NUON technology. Once again the game and the visualiser shared a lot of technology.

In more recent years we have been developing a visual synthesiser engine called Neon. Neon is a modular software synthesiser in the style of the many modular audio synths that have become popular in recent years, but operating on visual displays rather

than audio streams. The first generation of Neon was used to create the music visualisation engine in the OS of the Xbox 360.

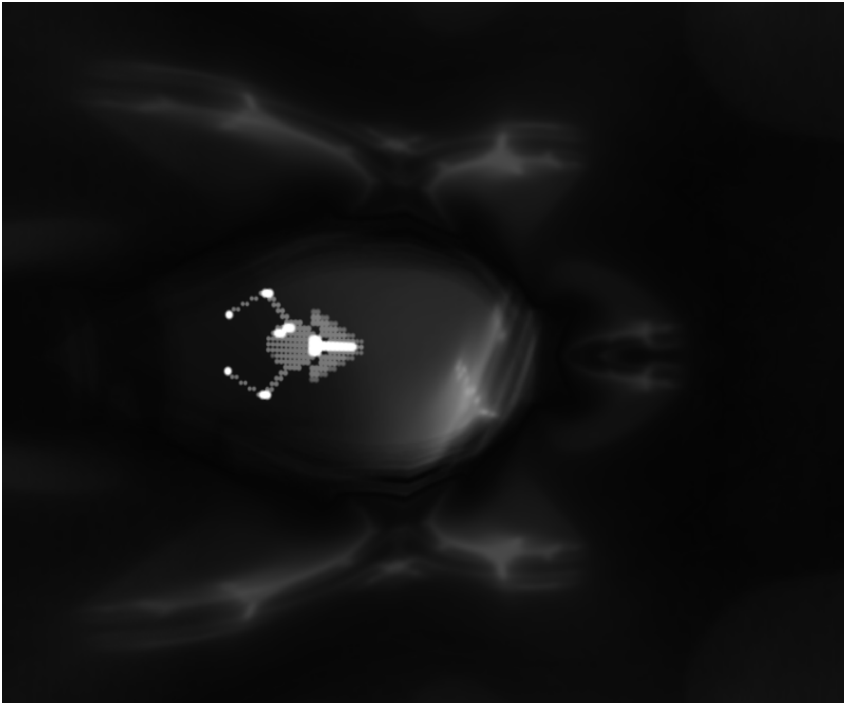
Neon is constantly evolving, and a version was developed that would allow for the construction of gameplay inside the synthesiser. This was used to develop "Space Giraffe", a game which probably blends the idea of abstract visual synthesis and gameplay together more than ever hitherto attempted. We learned from this that some people can get disoriented if you force them too far out of their gaming expectations and into a deeply abstract "reality". The game proved sharply divisive, between those who felt overwhelmed by what they perceived as an audiovisual overload, and those who could see how it all fit together and what we were trying to do. I received a couple of the worst reviews I've ever had from a couple of reviewers who fell firmly in the first camp, but I also received some of the best reviews I've ever had from others who "got it".



I've learned from this that we need to be careful to keep the gradient into deep abstraction from being too steep, lest we alienate people before they've taken their first step - but also that once people

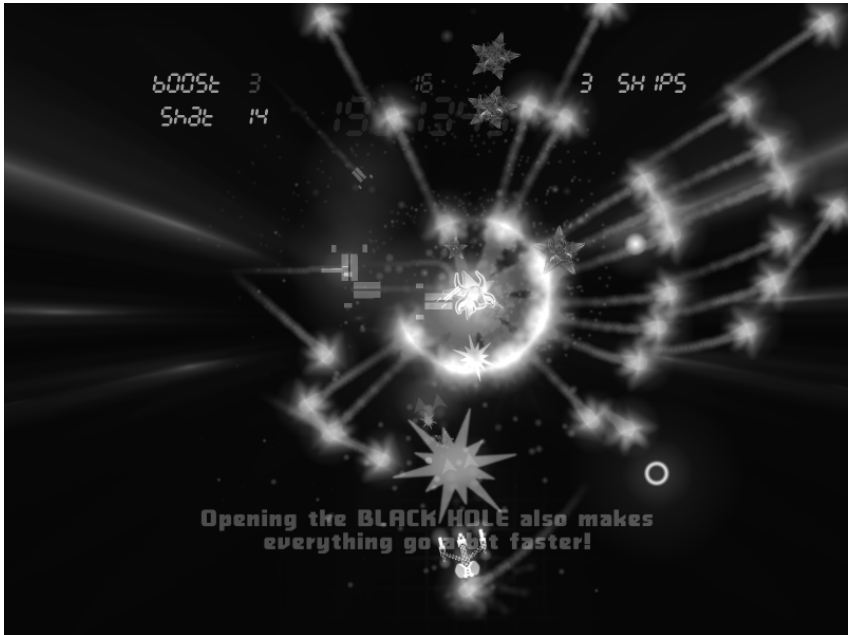
do take that step and get to where we wanted to take them, that they will find it immensely satisfying and enjoyable being there.

Neon has been evolving further and becoming more powerful - and with more power comes the capability for greater subtlety. It is becoming less dependent on conventional polygonal geometry and more shader-based, allowing for the generation of abstract displays of great complexity and beauty. I was very pleased when we were asked to develop Neon-based backgrounds for "Space Invaders Extreme" on the Xbox 360 - being asked to deploy our creation alongside such an iconic classic as Space Invaders was indeed an honour.



"Gridrunner Revolution" is built on top of the current and most powerful incarnation of Neon yet. There is less to overload the player than there was in Space Giraffe - the backgrounds are a subtle

canvas for the gameplay rather than a hyper-intense visual blast from within which one must struggle to pick out the relevant details. There are direct references to the game's pixellated ancestors, sufficiently familiar that nobody should feel disoriented and lost, but nonetheless updated and enhanced to look beautiful even to the modern eye.



It doesn't consciously try to look like "Tron", or emulate vector graphics, or to do anything except answer the question "what happens if I use a visual synthesiser to build a game?" - a question to which there are an infinite number of answers, and one which Llamasoft intends to answer in its own distinctive way many times as its Neon technology evolves.

