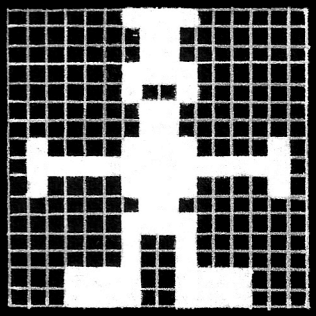
ROCK CRUSH



ZX81 16K

The Making Of a
ZX81 High Res Game...

Previously Unseen Original
Design Documents And Z80 Code!

Steven McDonald (2011)

Introduction:

Snake Oil For ZX81 Owners!

Snake Oil for ZX81 Owners!

You have to imagine the distrust when this full page advert began to suddenly appear in all the computer magazines circa 1984...

The ZX81 couldn't do hi-res graphics! Every owner of the machine knew that!

As we watched our school friends with their more expensive and modern Commodore and Spectrum computers, we were by this time only too painfully aware of the limitations of our beloved little door-stoppers.

Even the infamous wobbly 16k Ram-pack of added memory storage couldn't save us from this predicament!

The initial advert only featured Forty Niner but as time marched on, so did the second and third hi-res releases..

Surely these screenshots were faked or mock-ups?

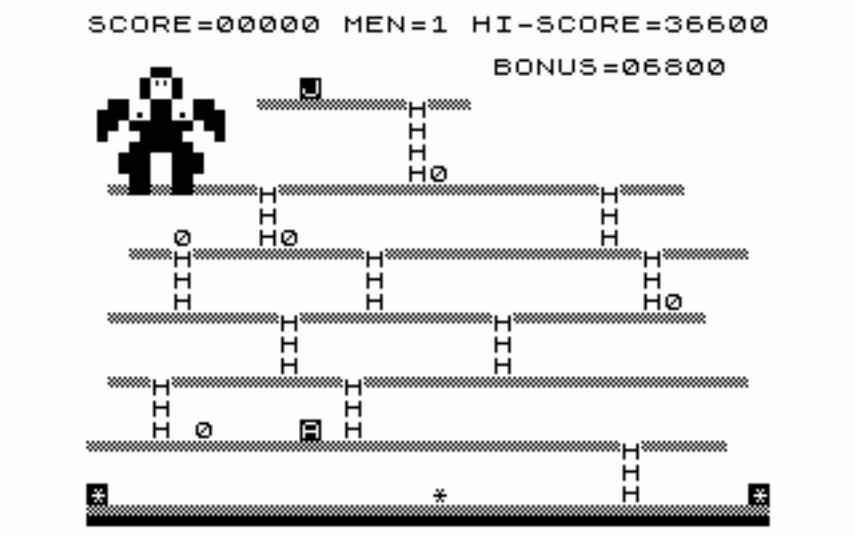
As time passed, we thought there had to be something in this after all.

Time to place an order!

You have to see what we were used to first...

Snake Oil for ZX81 Owners!

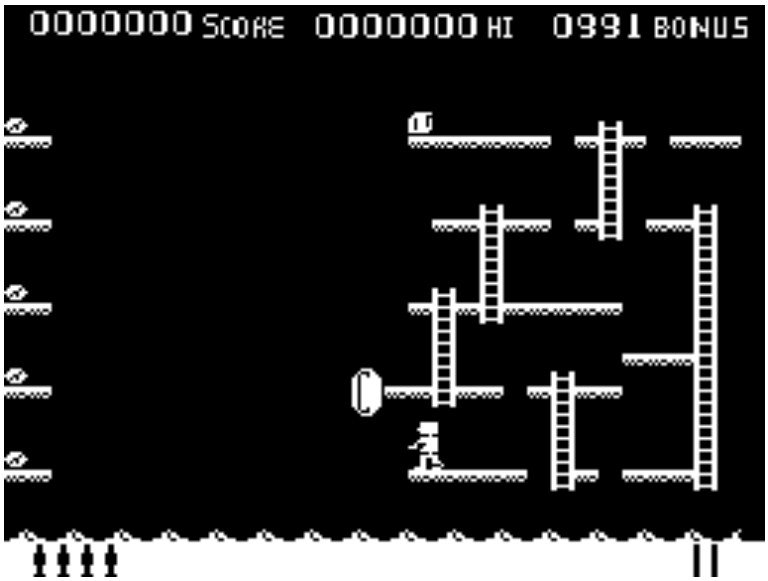
We knew when we loaded up Crazy Kong, we were going to be treated to a screen like this:



Yes thats right, our intrepid hero (here the inverse ‘A’ character standing on the second platform up), is about to run into some nasty rolling 0’s (sorry barrels!) and Juliet (the inverse ‘J’) at the top of the screen is screaming out to be rescued!

Exciting stuff! You get the idea.. ZX81 games up until this point were all text based. Letters A-Z and some punctuation signs running about the screen!

Boy did your imagination need to work hard to fill in the rest!



Anyway, back to Software Farm.

Boot up Rocketman and after an amazing splash screen of the cosmic cockerel, you are greeted to gorgeous graphics like these..

A site for sore eyes to ZX81 users everywhere...

Truly revolutionary!

Spectrum quality display resolution without any additional hardware!

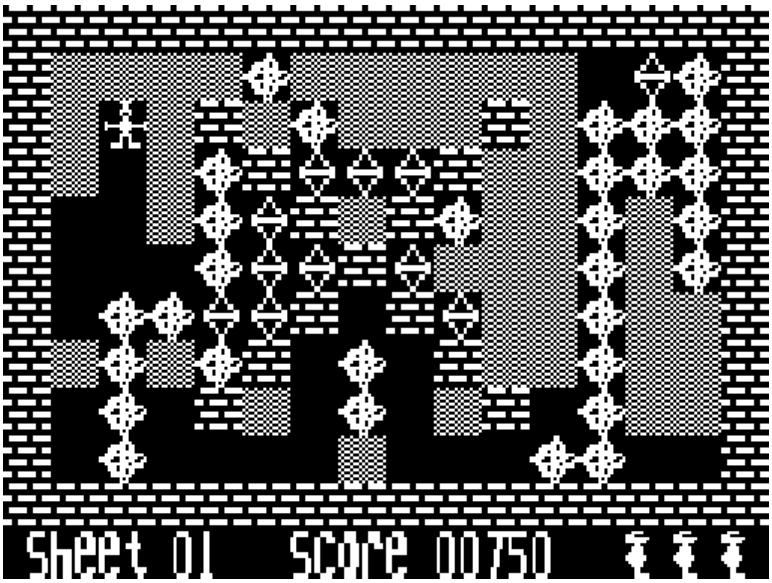
The overall effect was mind blowing!
Full 256 pixel across by 192 pixel down hi-res graphics!

Why all these references to Snake Oil? Was it too good to be true..

Well.. Yes.. and No!

Snake Oil for ZX81 Owners!

One things for sure...



If it hadn't been for me seeing Rocketman with my own eyes, Rock Crush would never have been made or indeed possible!

As soon as I saw it, I started breaking into it and tearing it apart - desperate to find its secret sauce so to speak.

Just HOW did it do those hi-res graphics with no additional hardware?

I worked day and night to find the answer..

Remember, there was no internet back then...

You had to conduct your own research and make your own conclusions as to what was going on...

I started learning machine code - the language these high res games were written in.

Eventually after literally poking around the screen display enough and disassembling the code called from the RAND USR line in the basic loader, I found the answers I was looking for...

Software farm had written their own display routine and were forcing the ZX81's hardware to work against its original design.

You see, the ZX81 would use its main processor - the Zilog Z80 chip to actually generate the picture on the users TV - along with the custom Ferranti ULA chip of course.
In fact, it was only in the gap at the top and bottom of each TV frame where it was free to run the users program - BASIC or machine code! That CPU was busy!
It was an ingenious and cheap way to make the chips do more together than they should have.

By very carefully taking control of this timing sequence and reprogramming it in a very clever way, pseudo high-res was possible.

Pseudo - in that you did indeed have 192 lines at your disposal down the screen.

Horizontally however, you did not have complete control of the 256 pixels across.
By complete control what I mean is it was not possible to program individual dots or pixels to come on or off at your command. Instead each bank of 8 dots (or byte in display memory), could only hold certain patterns of dots (predetermined by setting the Z80's I register).

Snake Oil for ZX81 Owners!

So it was snake oil!!!

Well.. whatever, the overall effect was amazing!
The bottom line was when designing graphics for this system, you could not simply say light that pixel and the one next to it but make sure the next three are off.

Instead it was more like pattern matching - you went through the sheets of patterns available and chose the pattern closest to what you ideally wanted.

Take another look sometime at Rocketman's graphics! Sure the man can jump up and down perfectly smoothly, but move him left or right - see those jagged pixels? Look a bit strange? Thats the effect of having to compromise on the patterns available.

Pixels across the screen horizontally are programmed in bytes - values 0-255 for the 8 bits. Sadly, these 256 combinations of dots are not available - instead its 64 patterns (seemingly 'random' of which some repeat - further reducing the count) plus their inverses. You added 128 to get the inverse. It turns out these random patterns are actually reading values in the Sinclair ROM chip but that's for another discussion.

Anyway, with all this worked out and the hi-res core routine identified and disassembled, the design of our first hi-res game could commence.

Myself and Scott Dolan worked for many months on a game design called Crazy Castle.

We painstakingly created the graphics and coded demo's of it in BASIC - that is we used our machine code hi-res core routine but once the new interrupt driven display driver was in use - we quit back to BASIC and poke'd all the graphics onto the screen. We used basic as it was quicker to develop and test with. It was also very slow..

It became clear after a few months that Crazy Castle was becoming very complex and BASIC just couldn't cut it for placing hi-res graphics on the screen - even for demo's. Our scenes were taking ages to draw up and animation was out of the question. It didn't help that each screen needed custom programming - all time consuming.

My machine code skills were coming along and I could write simple routines - but a whole game in machine code? I was untested and had doubts!

I secretly decided to call a halt and begin work on a new game with the idea of going back to Crazy Castle later. Unfortunately, we never did finish that first game off!

Anyway, long story short, I worked myself for many weeks to design and finish Rock Crush!

It was influenced hugely by the hit Commodore 64 game Boulder Dash and also on an advert I had seen of a game called Diamonds by Simon Hunt for the Atari (which I have never played).

Snake Oil for ZX81 Owners!

I can still remember to this day, firing up the whole machine code program for the fist time - all hand written in a notepad but totally untested.. You know what... It worked! First time!

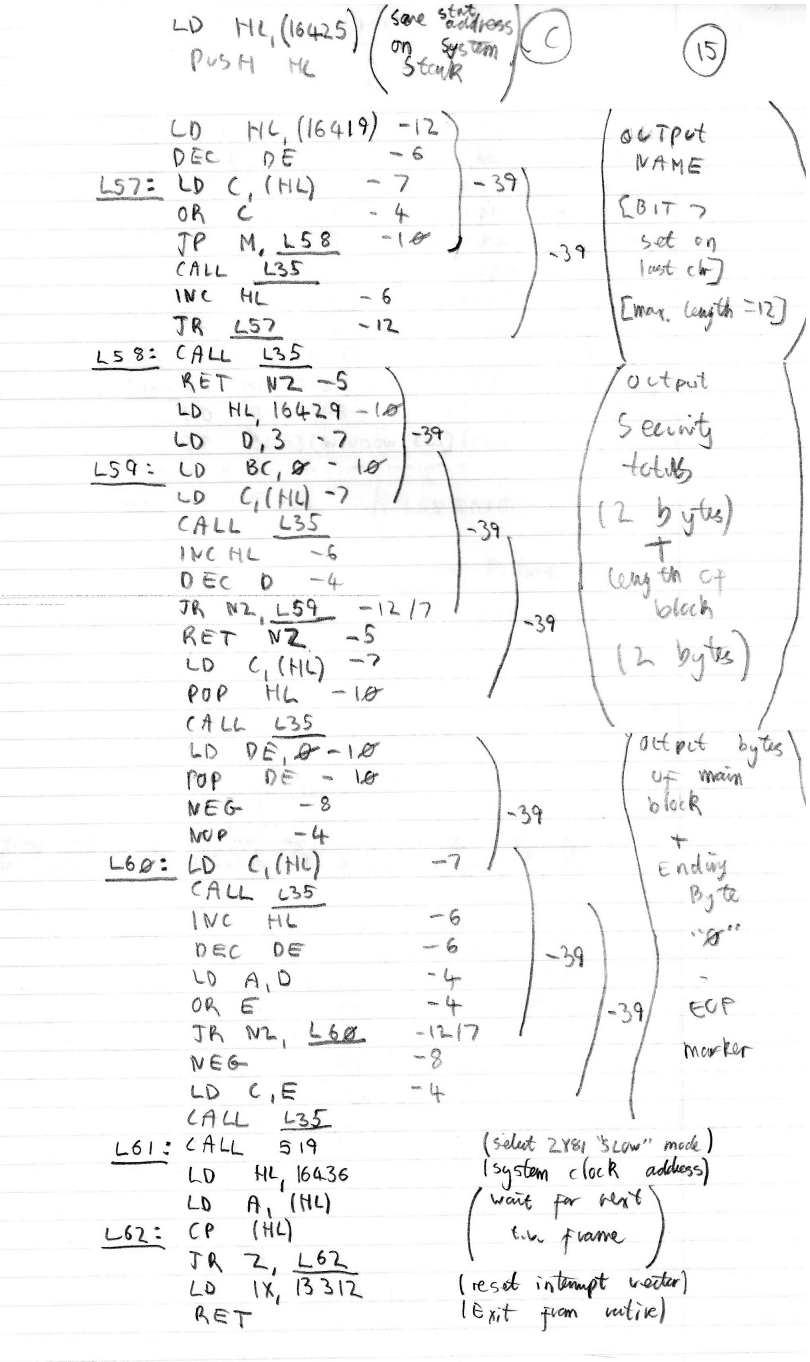
I think I had to alter the delays but other than that it worked!
Mind you, it crashed once you finished the first screen but that was because that was all I had designed and entered at this point - one screen! The engine however, expected more!

I showed Scott and he was amazed! He got to work immediately on the remaining 9 screens required for release along with the splash intro screen we jointly designed.

Later, we worked on the follow up - Dan's Revenge and sold both games worldwide.

We sent letters out to contact ZX81 owners and managed to make a profit eventually.

I should say that looking back - it's not my best Z80 work by a long shot.



I was new and pretty inefficient in my coding style and use of registers. Having said that, the game spent approx. 95% of its time running a delay loop just to slow the action down! Such is the power of machine code!

I wrote an English like pseudo-code of the main game logic and pretty much translated that line by line into the many machine code equivalent instructions to get that part done.

This approach helped me later when I rewrote the game in the Blitz language in 2001 - I just followed the same approach and hey presto - the game was now native to Windows!

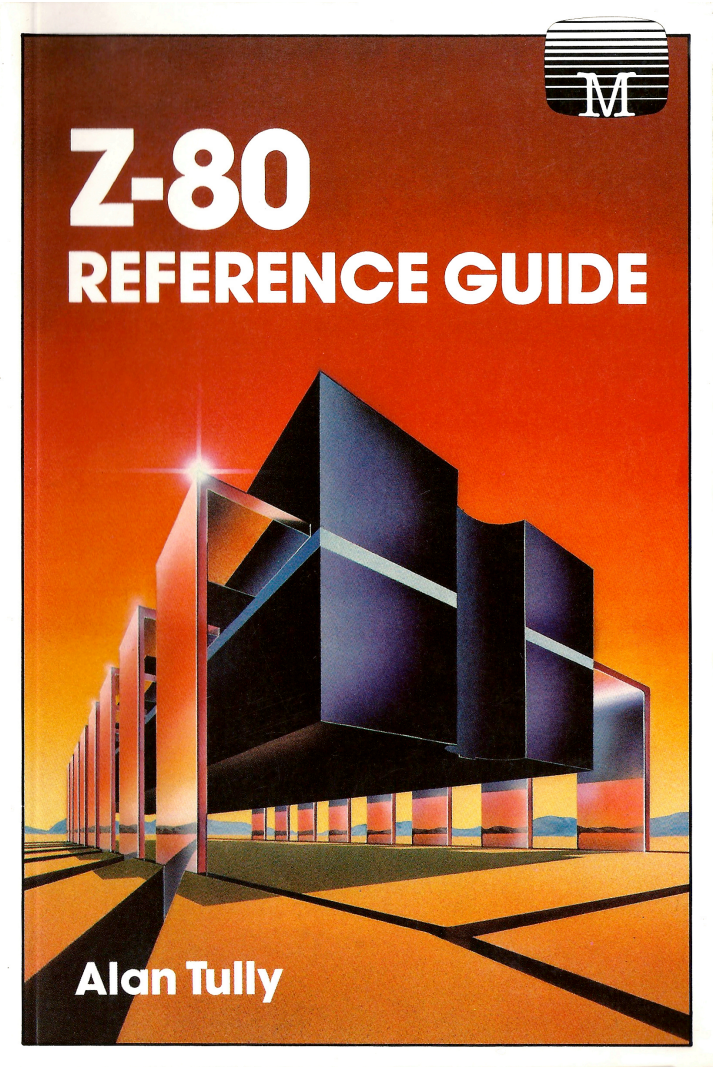
I started to develop a completely new replacement Operating System for the ZX81 in 1988 and to this day, this contains the hardest programming I have ever done.

In part of it, I was writing time dependent machine code - talking to a cassette recorder at 15 times normal speed - interrupts were off and I had complete control of the processor.

I had to make sure that no matter what branch of the code was jumped to or what I was doing, each byte output to tape had to be timed down to T-State level.

Snake Oil for ZX81 Owners!

I had 39 T-States between each byte - just being out by one T-State would compromise the timing of the signal. This applied to both Loading And Saving.



Each machine code opcode takes a certain amount of time to process in the Z80 processor. I had to know these timings and even write some dummy instructions just to make my timing spot on.
Get it wrong by just one T-State and I wouldn't be outputting a perfectly symmetrical square wave to the cassette recorder interface.
You can see above how I had to time each branch to 39 T-States... That was hard!
By comparison, Rock Crush, 4 years earlier, was easy!

(The best book I ever read on the Z80 processor was the Z-80 Reference Guide By Alan Tully, Melbourne House, 1985. It featured each Z80 machine code instruction, its affect on the Flags register and crucially, T-State timing information for each instruction!)

(One day I will get around to scanning in the whole ROS - Ramdisk Operating System documentation and making it available).

Snake Oil for ZX81 Owners!

As mentioned earlier, the original pseudo hi-res core routine was written by Julian Chappell of Software Farm, sometime around late 1982. I only later modified a few instructions in the internal core timing loop in my version. I also did some work on making the change smoother from hi-res to lo-res and vice versa.

There also appeared a pseudo hi-res toolkit by Richard Taylor from CRL and another extremely rare commercial pseudo hi-res game from Lothlorien Software - Micro Mouse Goes Debugging.

I disassembled both of these a few years later and both used Julian's core routine.

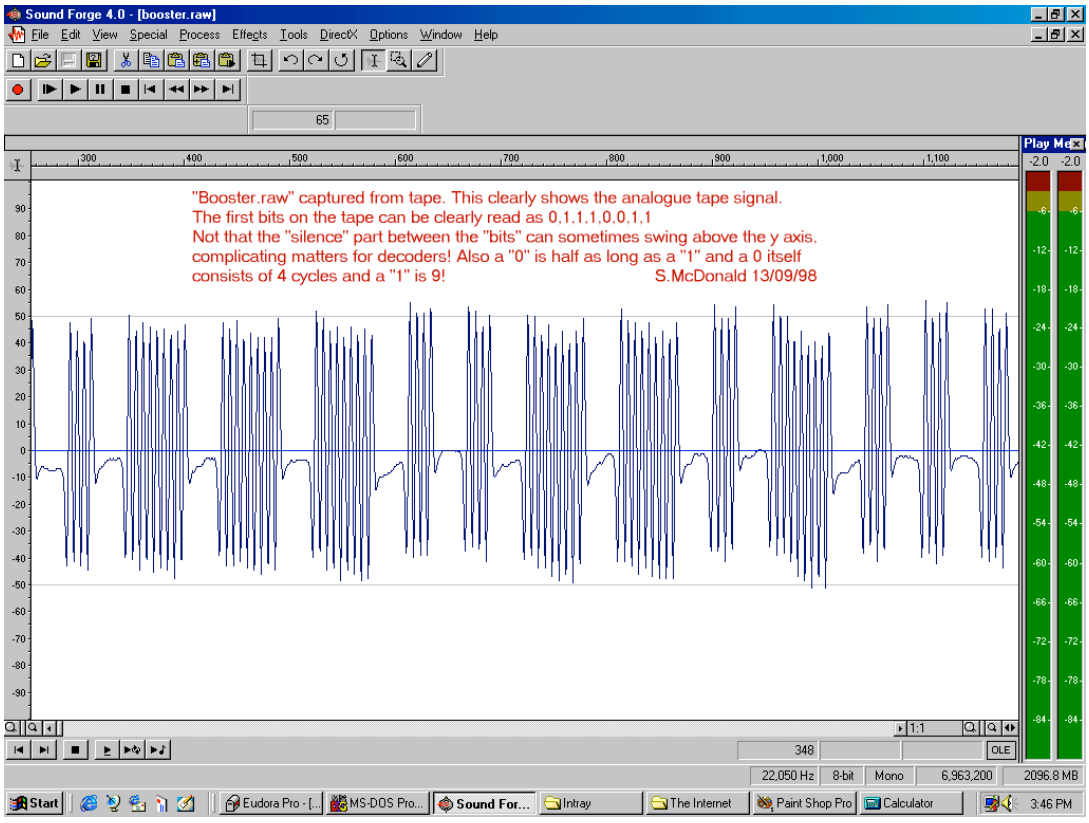
His background of industrial machine code programming of various CPU's made him ideally suited to the task of stretching the original design of the ZX81's TV display generation.

Incidentally, true hi-res was indeed possible on the ZX81, as long as you had the appropriate add-on hardware. The problem was - most people did not.
That being so, pseudo hi-res was the ideal compromise and could easily fool people.

Hats off also therefore to Wilf Rigter for his ingenious WRX true hi-res core routine which I did later do some programming for. For those interested - do a google search for "ZX81 video display" and read his extremely detailed description of all the different display systems invented for both true and pseudo hi-res. He really does know his stuff!

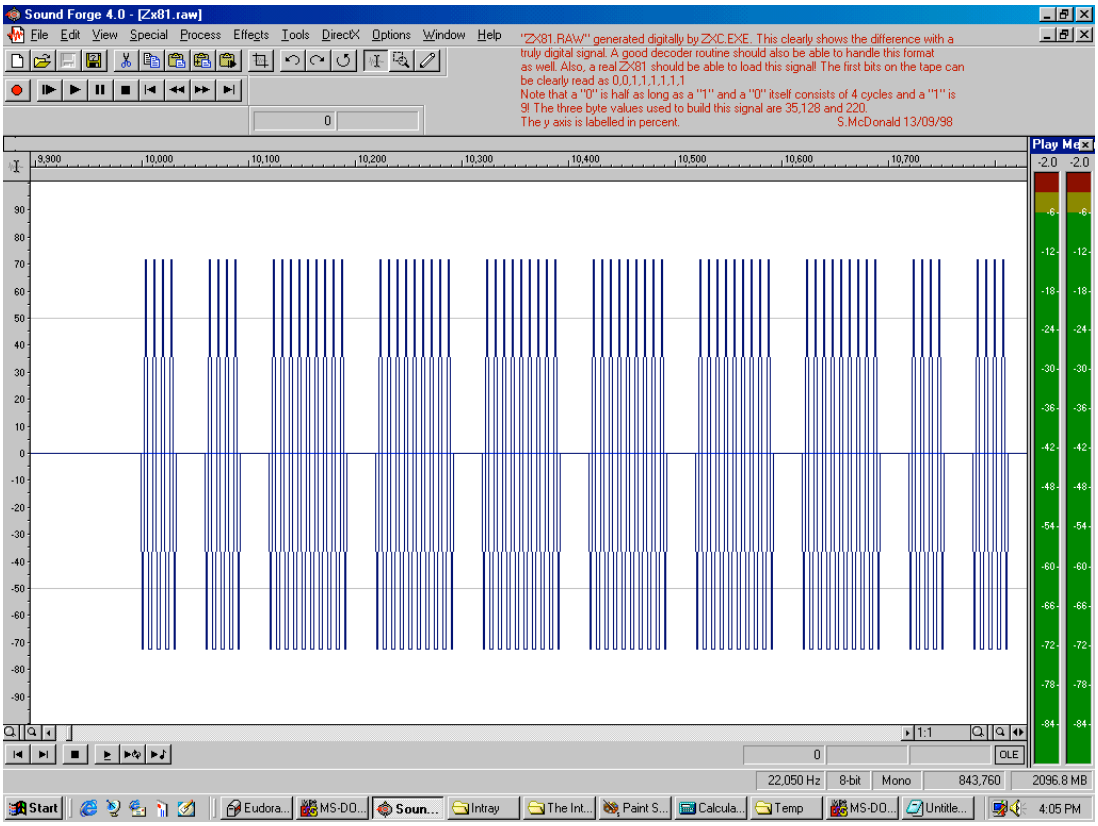
Anyway, time moved on, and slowly we forgot the ZX81...

It wasn't until 1996 and the arrival of Dial Up Internet and ZX81 emulators, that I went back to my old code..



I dusted off the original tapes and wrote some conversion software that allowed me to sample the cassette tape, digitize it and decode it and as a result managed to get the code over to the PC ready for the ZX81 emulators.

Snake Oil for ZX81 Owners!



I wrote both Decoding and Encoding software - ZXC.EXE and ZXD.EXE - both run on the PC and are available free from my website.

Once I had the code running again in the emulators, I had to make a few timing changes to Dans Revenge - the delays - so that under emulation they would perform as I expected. This was largely as a result of the underpowered machines of the time. Today this is not an issue and the original code can be run as intended.

Enter 2001...

I decided to revisit both my old games and bring them natively over to Windows.

As I said, earlier, I used the original English like pseudo-code and rewrote each game from scratch.

10 more levels were added to each game, along with a new soundtrack (borrowing from the C64 Boulder Dash game!), new sound effects and a new colour graphics mode!

The original silent black and white ZX81 hi-res mode was of course also available as the main default.

Now that 2011 is here, yet again I find myself tinkering with the old code, design, graphics and Apple's iOS SDK!

Who knows what's next...

Snake Oil for ZX81 Owners!

This book contains all the original design elements from the first sketch ideas of what the game would be based on - to the Z80 hand written machine code - to later versions, I have just talked about.

I can only apologize for the lack of comments anywhere in my code - years later it is a nightmare even for me to follow. All I can say is that as time moved on, I quickly learned my lesson and all my later machine code programs were well documented!

I can remember, I loaded the rocks and diamonds directly into the screen memory using pre-calculated absolute addresses for where each should appear.

Dans Revenge used a much more efficient method of encoding its screen layout!
Rock Crush was rough and dirty by comparison, but it got the job done!

Also, whenever I was moving Dan around the screen and needed to see what was above him or next to him, I would directly read into the screen buffer to see. I identified each type of graphic e.g. a rock or a diamond by looking at a specific row value that I chose so that I got a different value back when I looked at that same row for each different graphic.

There are some references to the second follow-up game, Dans Revenge, which was written later in 1986 but some of the design elements were similar so the same design books were sometimes used.

Why now?

Well, sadly a lot of the original design books and coding books are falling apart!

Recently, I realised it was time to scan them in and document them before it was too late.

I hope you have enjoyed reading this introduction and find this book of some use or interest.

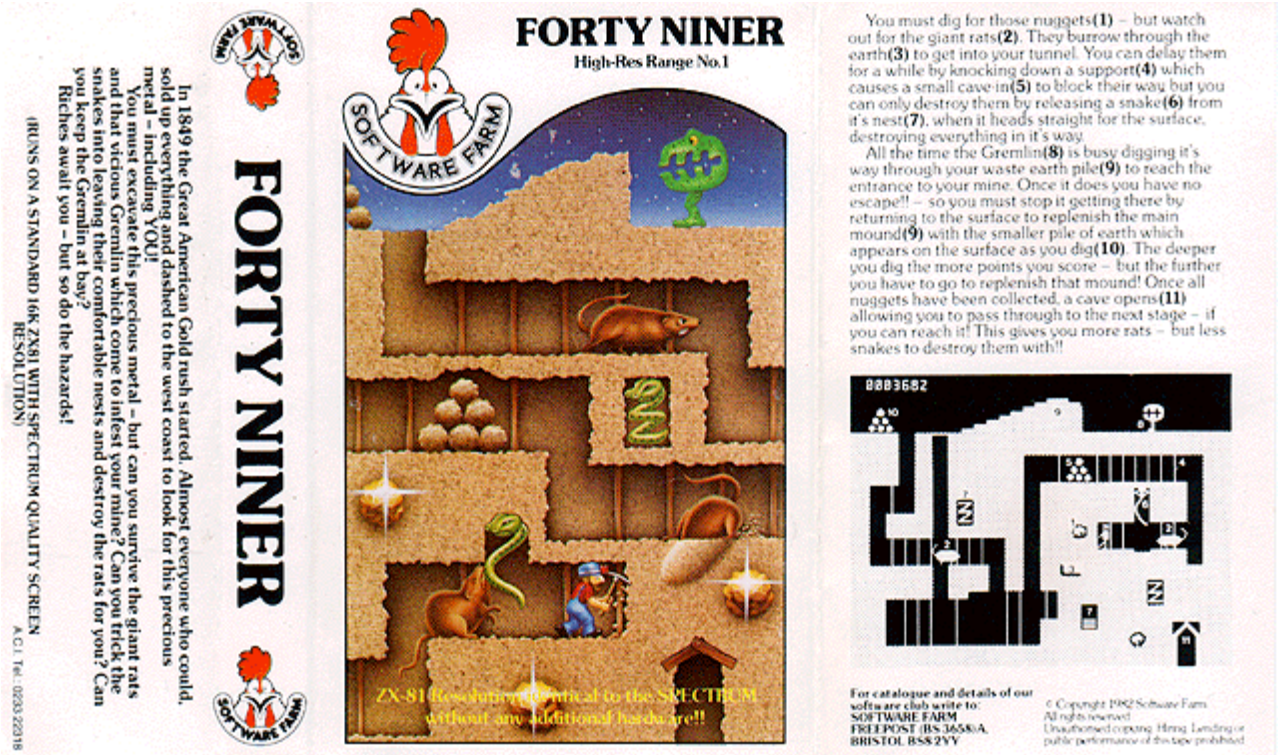
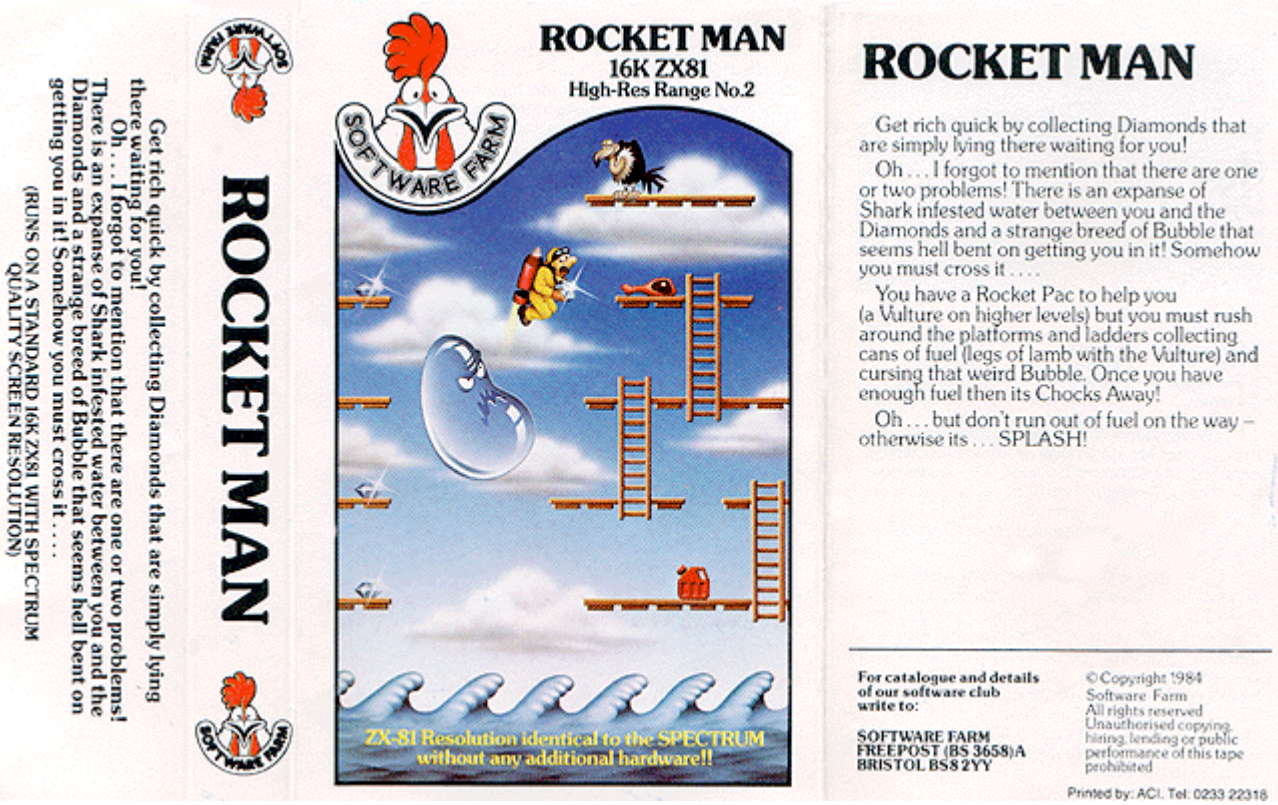
I have left these documents as they were originally as they are historical - as such some contact details and email addresses etc. are no longer valid. See below for up to date information.

Thank You for reading.

Steven McDonald
17th Feb, 2011

Contact: desktop@stevenmcdonald.net

Snake Oil for ZX81 Owners!

Cover Inspiration!

To this day, I have never seen better cassette game covers than these: