January 2018

Issue 20

THE SPECTRUM SHOW

Magazine

INFOCOM

GAMES ON YOUR SPECCY

FLASHBACK 87

GAME REVIEWS

HARDWARE

SPECIAL FEATURES

MIND YOUR LANGUAGE

Programming languages continued...

Includes material not in the video show!

LOCK & LOAD

Loaders and protection systems.

CONTENTS

12. INFOCOM GAMES
On your Spectrum.

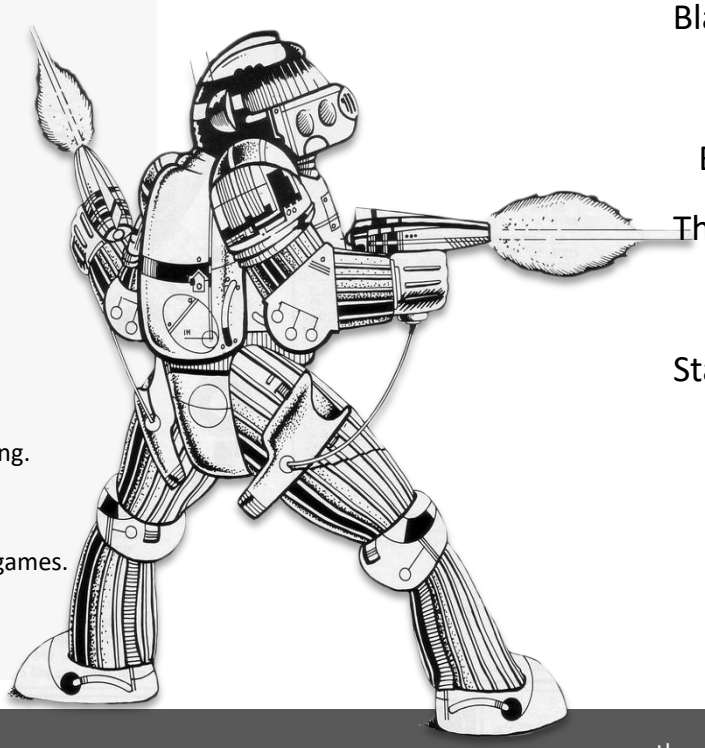
16. OCP ART STUDIO
Professional art package.

22. MIND YOUR LANGUAGE
Spectrum languages.

FEATURES

And more....

GAME REVIEWS



EDITORIAL

Welcome to issue 20 and thank you for taking the time to download and read it.

A new year begins and there is a lot happening in TSS towers at the moment. I have a new game out and two of my other games are now available to buy on real media. Yes, you can now buy four of my games on cassette just like in the old days.

Thank You To...

The great cover art for the games was produced by Kevin McGrorty, and my thanks go to him for doing such a fantastic job. Thanks also must go to Cronosoft for continuing to release games for many 8 bit machines in the original format.

This issue sees the end of a long running feature, Spectrum Means Colour, and I really appreciate the work Poitre has put into writing it.

That of course means there is a large hole to fill, so anyone who fancies writing something, please get in touch. You can write a one-off review, or a series of features on a specific topic. Just remember, your work will be archived on the internet and be available to read for years to come. Make your mark now, go on, give it go.

I am also beginning to plan the next series of the show, working out what hardware to review, what games to play and what to chat about with Geoff. I have enough hardware for the next series, but may run short later on. I continue to check online sites, but with ever-increasing prices, the cost becomes too much. This is one reason you can now support me via Patreon. (www.patreon.com/thespectrumshow).

LOAD Film 2

The second of Andy Remic's films was funded on Kickstarter, and now the hard work begins to get this film made. Chris Wilkins from Fusion Retro Books is also involved and together they have promised great things.

They will expand greatly on the first film, incorporating hardware, companies, magazines and even the modern games creation scene. That all sounds fantastic.

I have been asked to participate in the film, but as yet I am not sure which elements they want me to be involved in. I have tons of hardware, a long history of the Spectrum and of course my games. So those who backed the film will get to see me on the other side of the camera for once.

From one success to another...

Crash!

Those of you who backed the Kickstarter campaign for the new Crash magazine will have already received your edition, and a fine thing it is too. Chris has done a great job in preserving the look and feel of the original, and it was such a good read.

I hoped you liked my "Making Of…" piece. I had to cut it down a bit to fit the requirements of the mag!

The New Year

What have we to look forward to this year?

There is the Retro Revival show in Birmingham on May 19/20th and Play Blackpool on February 10/11th. Both events I hope to attend, and get my hands on those arcade games as well as getting to some of the great talks that are planned.

The three main UK shows are a credit to those that organise them and to those that attend. Together we are keeping the retro home computer scene interesting and exciting.

If you see a bald old man walking about with a camcorder, it will probably be me.

Fancy writing a game review or special feature?

I am always looking for new content and all contributions are welcome.

CLIVE RETURNS

Clive Sinclair returns to the computer market with his much anticipated portable machine, the Z88.

It is not what users were hoping for because it is not Spectrum compatible. This is due to the Amstrad buy out which means that Spectrum technology cannot be used. It also does not include, as many hoped, Clive's small screen technology used in the Pocket TV.

It does have an LCD screen supporting 80 columns with just eight lines, and from initial views it looks like a business machine, or as Clive labels it, a portable personal computer.

Gamers though will not be looking at this as an upgrade from the Spectrum.

SPECTRUM DISC

Amstrad have announced that a disc version of the Spectrum +2 machine will be released in the summer. Named the Spectrum Plus 3, the machine will retain the 128k of RAM and the keyboard of the +2, but will replace the tape deck with a 3inch disc drive similar to the one used in the Amstrad CPC range.

Up until now, there has been no standard disc drive for the Spectrum, with many third party suppliers releasing their own systems. This held back game companies, but with this move, Amstrad hope to see more disc based games by setting the standard.

MELBOURNE HOUSE SOLD

Melbourne House, the company that gave us Penetrator, The Hobbit and Mugsy have been bought by Mastertronic. The budget label has been looking to buy a full price company for a while now and it seems the time, and money were right for Melbourne House. The cost has been rumoured to be a 7 figure number.

Games will continue to be produced for each respective label.

ROTRONICS GONE

Rotronics, the company who produced the Wafa drive, has gone into liquidation. Despite efforts to sell the company as a going concern, no deal has been made. Rotronics say they will fulfil any orders and they still hope someone will take on their products.

ELITE DROPS LABEL

2.99, The budget label set up by Elite Systems has been put on hold due to the continual legal action by other companies who provided the software.

Steve Wilcox, it seems, has not paid royalties to many of the third party labels such as Vortex, causing a flurry of legal challenges. With no resolution in sight, it looks like 2.99 will fade away.

SEE US AT ZX MICROFAIR
YOUR CHANCE TO WIN AN MO184 KEYBOARD
CONTACT US FOR DETAILS

The MO184 for SPECTRUM users

Designed by Programmers and Repair Specialists for Simplicity, Speed, Safety & Space.

KEYS:
- ★ Low-profile Keys, only 4 mm travel.
- ★ Large keycaps.
- ★ Readable at 3 feet.
- ★ Bright colours for easy identification.
- ★ Strategically placed for easy
  - extended mode
  - delete
  - → ← ↑ ↓
- ★ Extra keys for left-handed users.
- ★ Full-size space bar.
- ★ Numeric pad with decimal point.
- ★ Double width enter keys.
- ★ Quiet action on gold contact switches.

HOUSING:
- ★ Glossy wipe - easy material.
- ★ Fully insulated no mains flying around.
- ★ Generous height and depth to House Interface 1 & power supply, WITHOUT REMOVING FROM CASING.
- ★ Large slots at rear for Spectrum and Interface 1.
- ★ Palm rest, to save tired wrists.
- ★ Pen holder.

ASSEMBLY:
- ★ No soldering required.
- ★ Full instructions supplied.

FURTHER EXPANSION
- ★ 3 slot (female) connector cable @ £12.50
- ★ 2 slot (female) connector cable @ £9.00
- ★ 8" Microdrive cable @ £5.50

DEALERS, CONTACT US FOR AN OFFER YOU CAN'T RESIST!

Please send _____ MO184 Professional Keyboard @ £54.95 each*
_____ 3 slot connector cable @ £12.50 each
_____ 2 slot connector cable @ £8.50 each
_____ 8" microdrive cable @ £5.50 each

I enclose cheque/P.O. value £_____*(add £2 p&p)

NAME .....

ADDRESS .....

.....

TEL: ..... POSTCODE .....

£5 OFF!! THE MO184 KEYBOARD IF YOUR ORDER IS RECEIVED ON THIS COUPON BY THE 10/JAN/85

Games programmers, watch out

3D TANX

DK'Tronics 1983

For me, this is one of the better 16K games and, to be honest, I thought it was a 48k classic. This game will probably be known to many Spectrum gamers, especially those who got their machine early on: it was one of the 'must have' titles that really shined in an otherwise infant market.

The idea is simple, stop the tanks from getting across the bridge. Sounds easy, but it isn't. To do this you have your own tank and you control the gun turret. You can move left and right, and raise or lower the gun. This dictates how far your shell will go when you fire.

The enemy tanks appear on the bridge in the distance, in one of four lanes. To hit them not only do you have to judge their movement and how long your shell will take to get there, but also how far your shell will travel based on the angle of your gun.

Hitting a tank first time will disable it, leaving on the bridge unable to move or fire back. Yes, the tanks do fire back, and this means you have to make sure their shells do not hit you.

You will soon learn that certain elevations will hit one of the four rows on the bridge and then it becomes easier to target though the game remains a challenge.

Disabling a tank means it stops other tanks from crossing the bridge, however, they can, and do shoot disabled tanks to move them out of the way.

You also have limited ammo, with replenishments arriving at intervals.

Sound is excellent, with some great firing and explosion sounds, although I was never a fan of the early Don Priestley

game menus with the constant beep while you change the settings.

The graphics are well drawn, detailed and move smoothly and this is a great little pick and play game that will be a favourite of many players.

Certainly worth a blast.

DECOR WRECKERS

Scorpion Software 1984

This simple little game is typical of the releases in the early years, between 1982 and 1983, and proves to be very addictive.

The idea is to paint all of the screen using a limited amount of paint by guiding your ever-moving roller. This in itself can be tricky as the roller does not stop and even when you paint over an existing area, it still uses paint.

Once all the screen is covered you guide the roller to the bottom left block to complete the level.

There are other things out to cause trouble though. The first level has what look like mice, randomly appearing on the screen. These have to be painted over to allow you to complete the level. The next level has a spider that moves downward, removing the paint as it goes. This means you then have to repaint it. Luckily, colliding with it does not kill you.

The graphics are simple but effective and move smoothly, and the sound consists of just clicks and beeps.

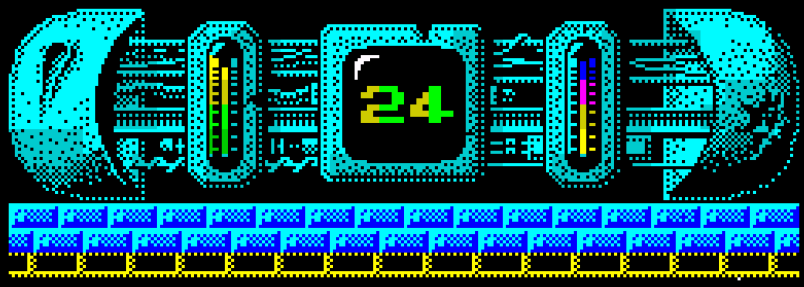
I like this simple game, it reminds me of A&F's painter, although unlike that game painting a square does automatically fill it in.

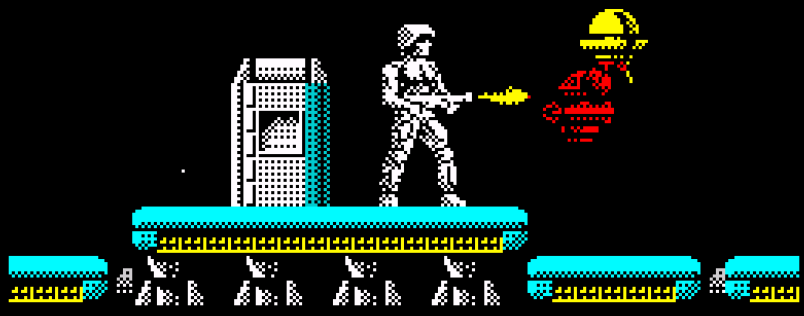
I managed to get to level 6 and discovered other obstacles including snails, that cleared the paint horizontally - a real pain!

Control is good and overall this is a nice little pick-up-and-play game that's worth a quick play.

TRANTOR THE LAST STORM TROOPER

Go! 1987





Trantor has been betrayed and left alone on an alien planet. There is no further details about this act, but the intro animation gives you some clue.

Left alone, it's now a race against time to activate the transporter so he can escape and presumably take revenge.

To escape, he has to run around the alien complex trying to locate 8 terminals. Each one holds a letter of the activation code. Of course it is never that easy, and there is a very tight time limit. This can be seen ticking down at the top of the screen but, to be honest, you are too busy trying not to get killed.

As you run around, there are various containers that when examined, will produce health, reset the timer or refill your weapon energy. To examine them, you just crouch down in front of them.

On to the game then and, as you can see, the graphics are very nice. Large, well animated and smooth. Trantor can run, crouch and jump, as well as fire his flamethrower.

There is always something to shoot but, because of the time limit, it is sometimes best to just run and hope you can keep finding health pick-ups.

There are lifts to other levels, where the backgrounds, also very detailed, change giving a variety of different areas.

Sound is used well, with some nice music on the start screen and good effects for various elements of the game.

As a game though, I am not too convinced. That time limit is a real killer meaning you have to learn the game map to survive, and even

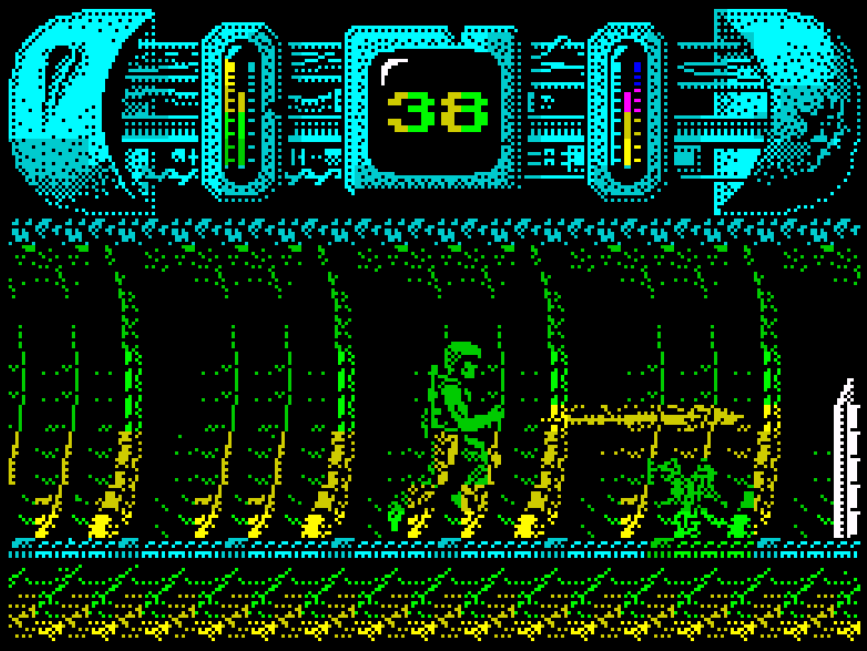
then it can be tricky. There is no time to stop and fire, you just have to find the next health pickup or terminal.

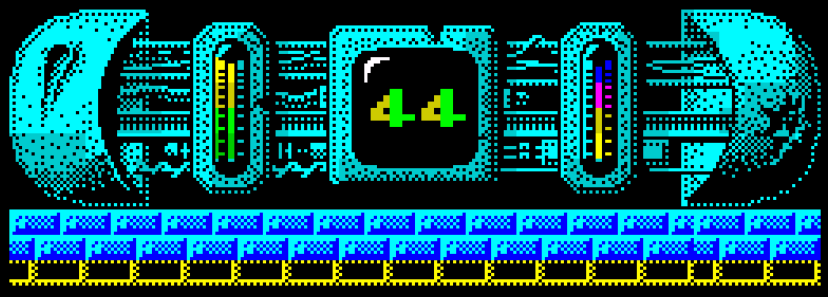
Once you have all eight letters, you head off in search of the transporter so you can enter the code and escape.

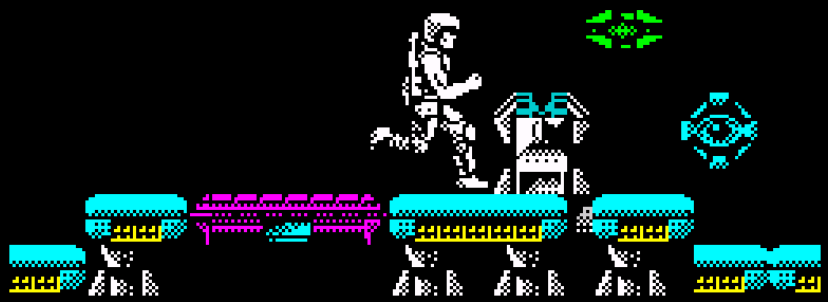
I never got all eight letters, but did have a fair few games that started off with me admiring the graphics only to run out of time. That soon edged towards frustration and the timer kept running out despite me hurtling about like a mad man!

Using a poke to stop the timer improved it slightly, as it then became a survival game, which certainly appealed more to me than rushing about trying to beat a clock.

A decent game then, and one that shows off what the Spectrum can do but just be prepared for a lot of frustration until you learn the game map.







KOSMIC PIRATE

Blaby Computer Games 1983

Many years have past since the Krell race have been ejected from their own planet of Verox. The Krells now roam the galaxy looking for anything they can lay their greedy little hands on. This time it happens to be supplies leaving a factory on the way to the warehouse. You must dive down between the fleet of ships that has been assembled to roam the planets surface and stop any unwelcome intruders.

Here then is a simple arcade game from the early life of the Spectrum, and to be honest, it shows.

At first this game reminds me of Lunar Rescue but the gameplay, although similar, is different enough for it not to be considered a clone. It definitely takes inspiration from the arcade game though.

My first attempts to get anywhere were terrible. The game is controlled by the cursor keys, which were always a bad idea, and this sent my ship to many a fiery death. Back in emulation and at least I could use the joystick option and this meant I could at least get a bit further.

You control a small craft and have to navigate to the bottom of the screen and pick up one of the boxes moving from left to right. Once collected you than have to carry it back up your ship. However, you have a limited amount of fuel to do this and this really does force you to make errors as you rush to complete the task.

Control is in character jumps, which makes things tricky, especially trying to get through the layers of space ships. If you wander too far off the screen, you explode, which is bit unfair considering the poor response to key presses. and you often find yourself being blown to bits for reaching the top of the screen.

The graphics are simple and lack depth or form and move in 8 pixel jumps, giving the game a type-in feel.

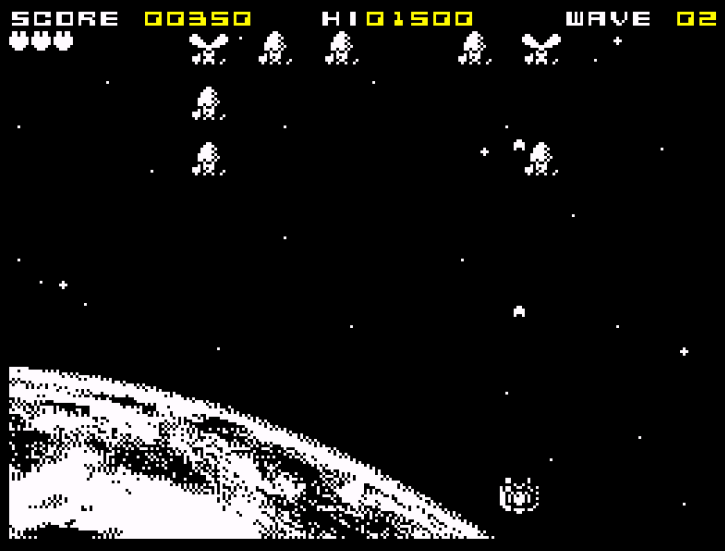
The sound is average for an early game, and has some nice effects here and there, mainly when you explode.

Gameplay soon gets boring as all too often your fuel runs out, with no means to refuel apart from getting one of the boxes back to your waiting mothership.

A simple game then, for a simple time in gaming. Some early games were ground breaking, others though were dull and repetitive… this one falls into the latter category…

BLACK·STAR

Juan J. Martine 2015



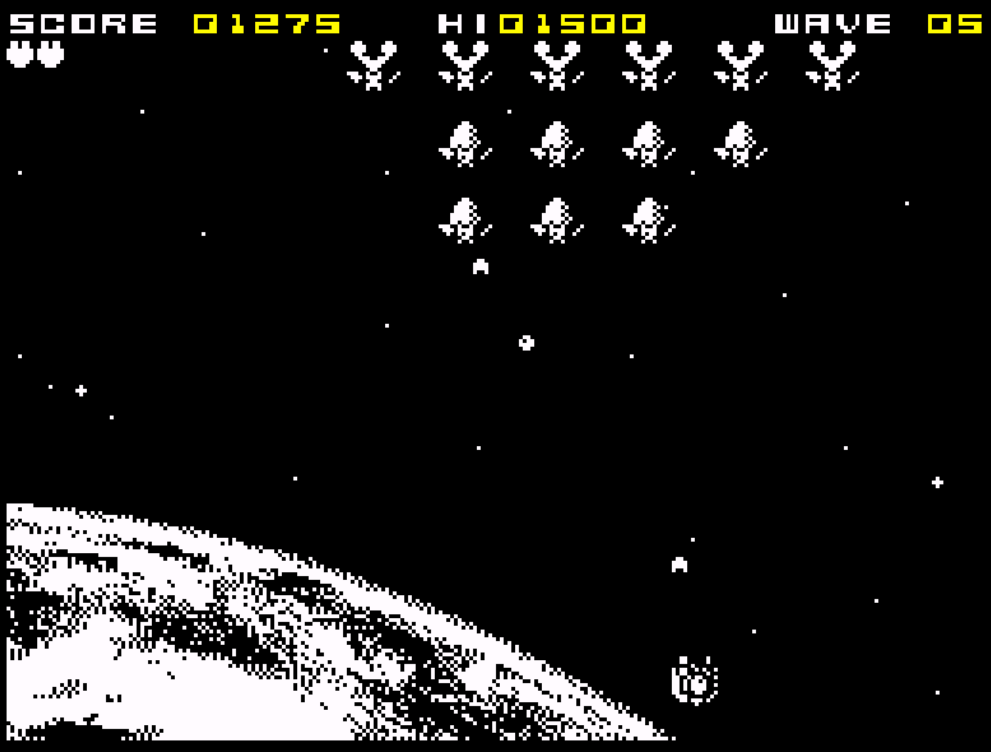
The solar system of the Kingdom of Heavens has enjoyed peace for thousands of years until an unknown force appeared in the sky of one of their worlds: a wormhole to a dying system. The last hope of the Kingdom is the battleship class Black Star, which will try to repel the horde at all costs.

This is the story for what is a great new shoot-em-up for the Spectrum. You can't beat a good game of Space Invaders, and Black Star certainly delivers.

Armies of aliens stomp across the screen just waiting to be blasted and, like the arcade game, as each one is shot, the rest get faster.

You have to be accurate to clear them all before they get too close.

The graphics are excellent. A planet takes up the lower left, adding a bit of atmosphere, and the monochrome aliens



are well drawn and animated and move smoothly. The player ship banks as you move left and right adding a bit of polish.

Sound is used well with a nice tune on the intro screen but in-game there is just the stomp and firing sound.

As each wave is cleared, a new wave appears in a different formation and firing more often. They get faster too, so it soon becomes challenging, but in a good way.

All of this makes for a great game and one to seek out if you are a fan of arcade shooters.

Highly recommended…

INFOCOM

ON YOUR SPECTRUM

Infocom produced some of the best text adventures ever to grace computers between 1977 and 1987. Other games were produced outside of these dates, and even after they were bought out by Activision in 1986, but their name is synonymous with quality.

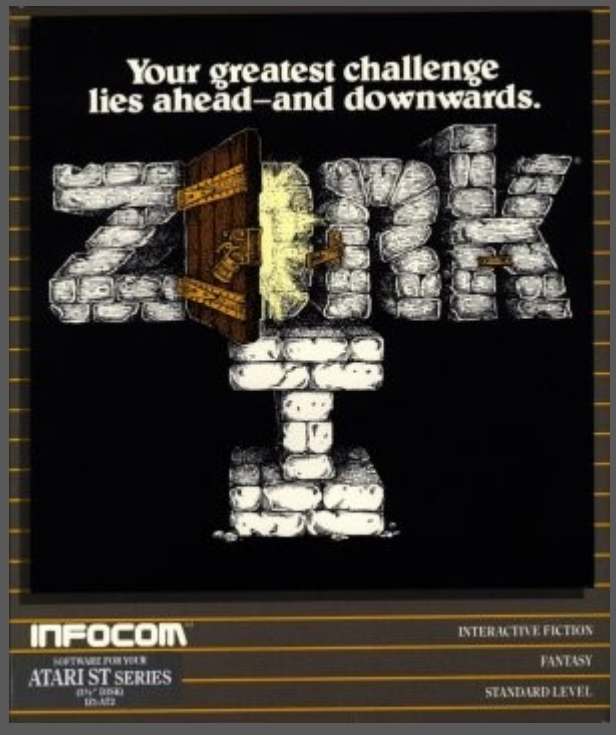
When Activision acquired the company and its games, there was a rumour that they would push out versions for the latest ZX Spectrum, the Plus 3. Infocom games required a lot of storage space that could be accessed fast, and cassettes just could not do the job. The Plus 3 though had a disc drive, and this was ideal.

Sadly this never came to fruition, leaving us Spectrum owners with just Level 9 and Magnetic Scrolls for our high-quality adventuring fix.

Infocom games were written by a tool named ZIL – Zork Implementation Language, which produced an output in byte-code. This code could then be used on any machine that had a Z-Machine interpreter. The idea being, the game was written once and to make it work on a particular computer, they just had to write an interpreter for it.

Z-Machine interpreters were available for most of the popular computers, and are still supported. Versions are available for PCs, Apple Mac, Amiga, Atari and even IOS and Android.

Around 2006 a Z-Machine program was written for the Spectrum by John Elliot that supported most of the Byte-Code versions, and using this players could finally get their hands on some Infocom games.



Zork 1: Inspired by Colossal Cave, was released in 1980 for the TRS80 computer. It sold over 1 million copies.

I have tried to use this tool many times, each time failing, but having a few days spare, I set about trying to finally get it to work and was successful.

Step 1 – CPCFS

Download and un-zip the contents into a folder on your hard drive. It is important to note that this tool will only run on 32bit operating systems. I had to use a virtual machine to get it working.

Once unzipped, there is nothing else to do at this stage.

Step 2 – The Game File

Once you have located the game of your choice, add it to the CPCFS folder. The game file must be a .Z5 format and must be version 3,4,5 or 8 of the Z-Machine byte code. You can find details about versions on the Infocom website. (http://www.infocom-if.org/index2.html)

Step 3 – Preparing the image

Using the command line, go to the CPCFS folder and enter:

NEW [name].dsk

For example:

NEW planet.dsk

This will create an empty disk image, ready formatted and ready to use.

To copy the game file to the image use:

PUT planet.z5

You now have to close the disk image using:

CLOSE

What You Need:

1. ZXZVM the virtual interpreter from John Elliot's page. (http://www.seasip.info/ZX/index.html)

2. CPCFS, a CPC file system emulator from anywhere you can find it. Currently WOS is broken and the link does not work.

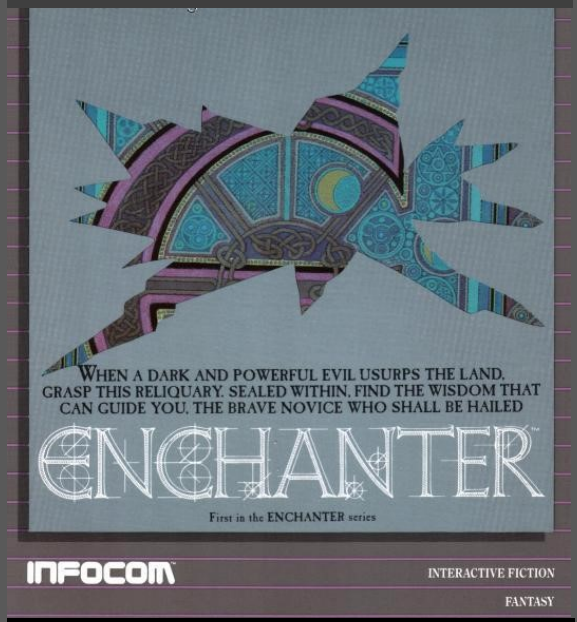
3. Infocom games files (.z5 format).

4. SAMdisc a tool to write .dsk images to real discs. From Simon Owen's site (http://simonowen.com/samdisk/)

5. A PC with an internal 3.5inch disc drive.

6. Some blanks discs.

7. A lot of time!!



Step 4 – The Z-Machine

Take the disk image you have just created and insert it into your Spectrum emulator. Do not try to load or run it. To check everything is ok, you can CAT the disc to see the file.

Add the ZXZVM.TAP to the emulator too, making sure the disc image is still there.

Load the TAP file and you will see the intro screen to ZXZVM.

First enter the drive letter:

A:

You will now be shown files on the disk – in this case just the planetfall file.

Enter the filename and press enter.

You are now asked if you want 32 or 64 characters. Make your choice and the game should load.

In the background, the required files have been copied to the disk.

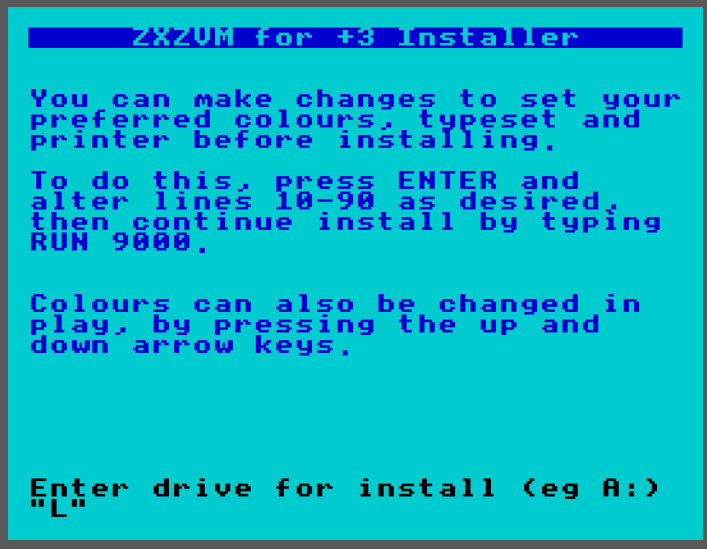
You can change background and foreground colours by using SHIFT 6 and SHIFT 7. You can also edit the loader that has been created on the disk.

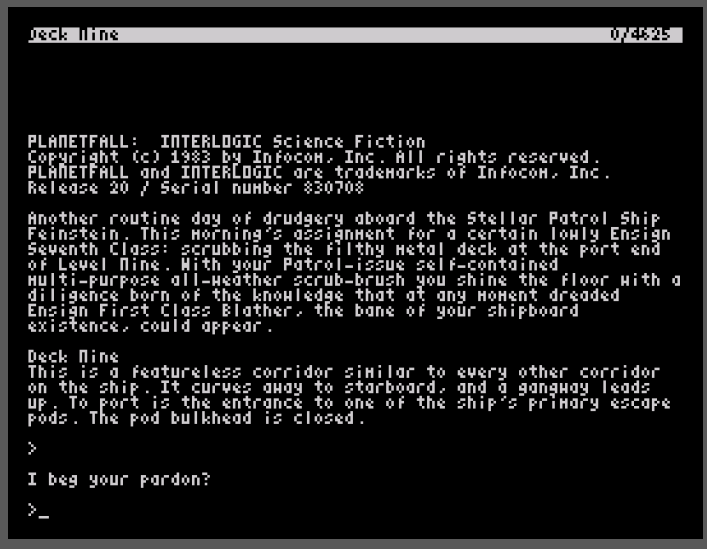
To play the game again, you just have to insert the disk image…

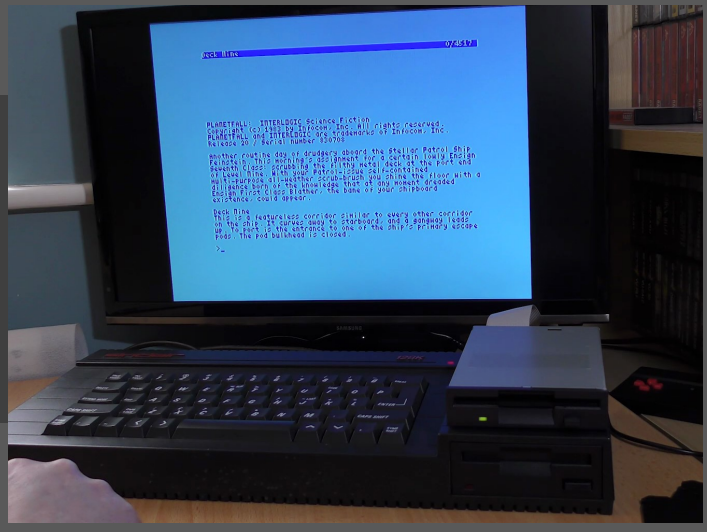
Enter LOAD “ZXZVM.BAS” and type in the file name.

This of course can be automated if you know a bit about Sinclair BASIC.

Repeating the process, and you can create as many disc images as you like, and play the majority of Infocom games.







Infocom Games: Legality

There are some games that can be downloaded free of charge, with hopefully more on the way. However, for most of them, it is illegal.

The best way to get these games is to seek out The Lost Treasures of Infocom compilations on eBay.

Step 5 – Real Discs

I have covered this before in Episode 17, but you need a PC with an internal 3.5inch drive and SAMdisc. The internal driver must be set in the BIOS to 360k 5.25 disc.

Extract the contents of the SAMdisc zip into a folder.

Copy the .DSK file into the same folder.

Using the command line enter

SAMdisc [image name] a:

For example:

SAMdisc planet.dsk a:

The data will now be written to the disk.

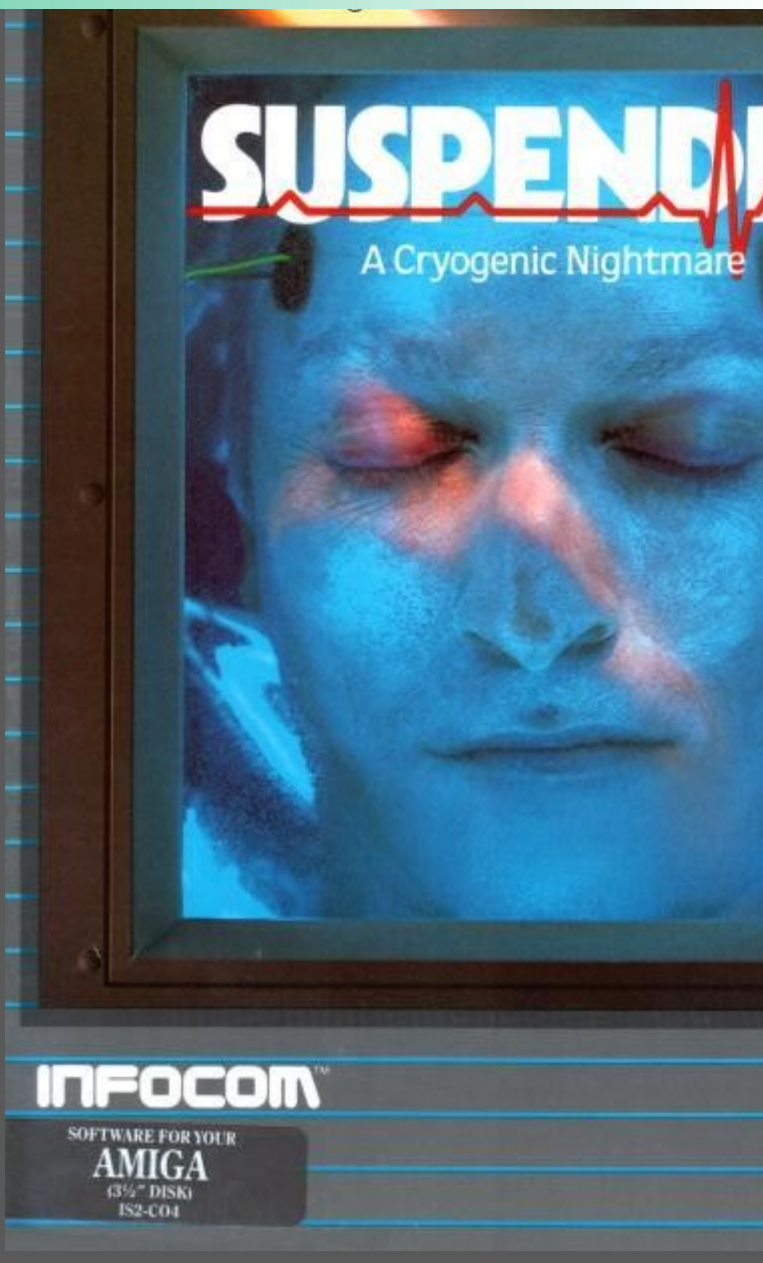
Step 6 – The Real Spectrum

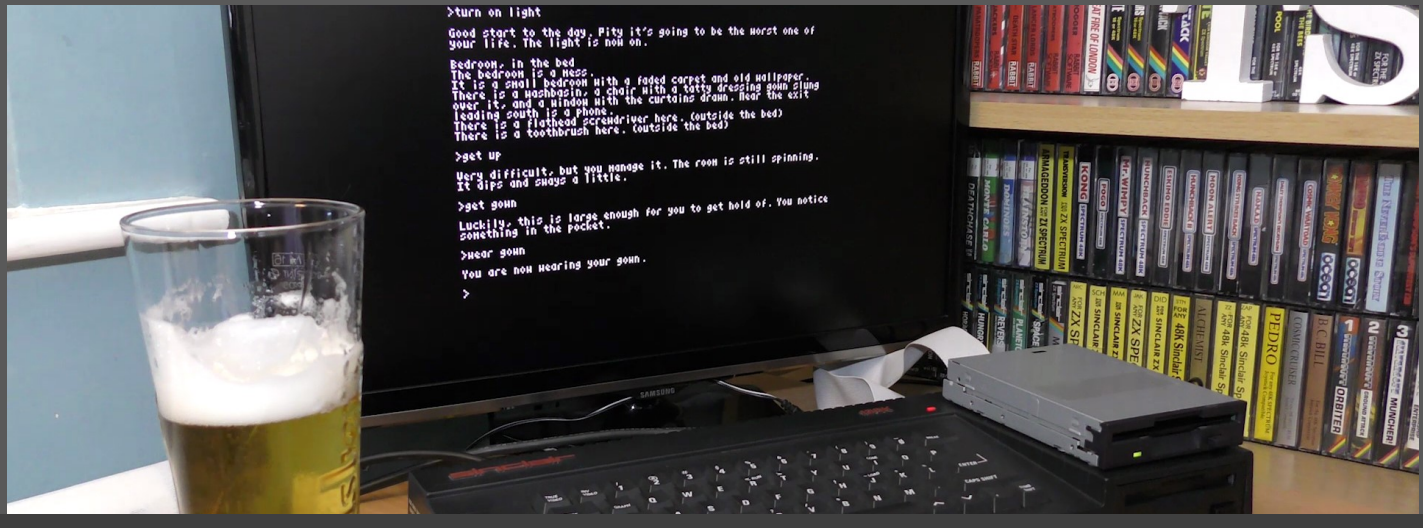
Once complete, remove the disc from the PC and insert it into the real Spectrum.

Using the same commands as before, you can now load the game.

And there it is. A real Infocom game working on a real Spectrum.

I have played Infocom games on many machines, so to get it onto the Spectrum, finally, is great. The 64-character mode takes a bit of reading, but it's a great feeling that the games eventually made it.





OCP ART STUDIO

The ultimate Spectrum art package?

Mouse controlled art package that borders on professional

OCP Art Studio was bundled with several peripherals including various mice, and was also sold as a standalone product. It was a solid, professional drawing program aimed at more than just the doodlers among us. The large and detailed manual covers the many features this tool has, so let's dive in and take a look.

For this review I will be setting the program up to work with a mouse. OCP Art Studio was designed to work this way and it is how you will get the best out of it.

The screen displays a menu at the top and a large mouse pointer hovers on-screen ready to start. The menus are familiar, and reflect many of today's art programs.

The Print menu; obviously allows printing if you have a printer attached. The File menu allows loading and saving as you would expect and the Attrs menu allows you to set ink, paper and border colours as well as other values.

The Paint menu allows you to select how you want to draw on screen, with sub options for more detailed settings such as brush shapes and widths.

The Misc section gives options to clear the screen or set the grid, which is useful when working with colour.

The Undo menu does what you would expect and lets you undo a mistake.

The Windows menu allows you to define areas on screen and then do various actions on them such as cut and paste, invert and rotate. This is a very powerful feature and lets you grab bits of screen, mirror them, pasted them somewhere else, scale them and generally do what you would do in a modern graphics package,.

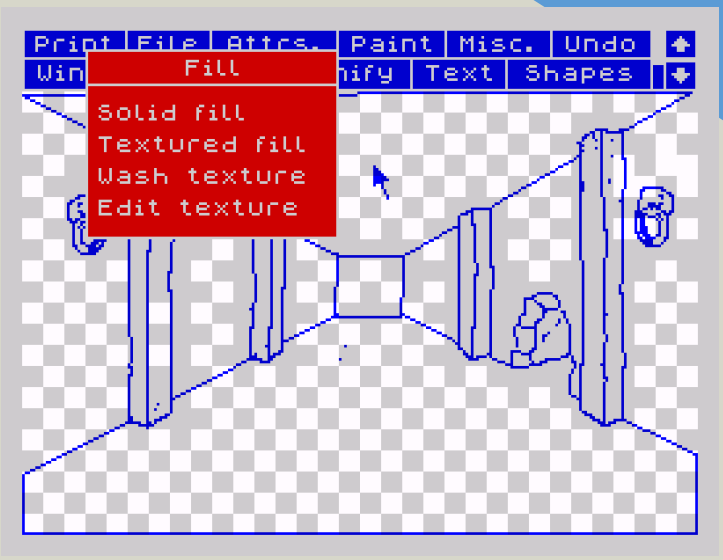
The Fill menu gives you various fill options along with sub options for fill types and patterns. You can even create your own fill patterns.

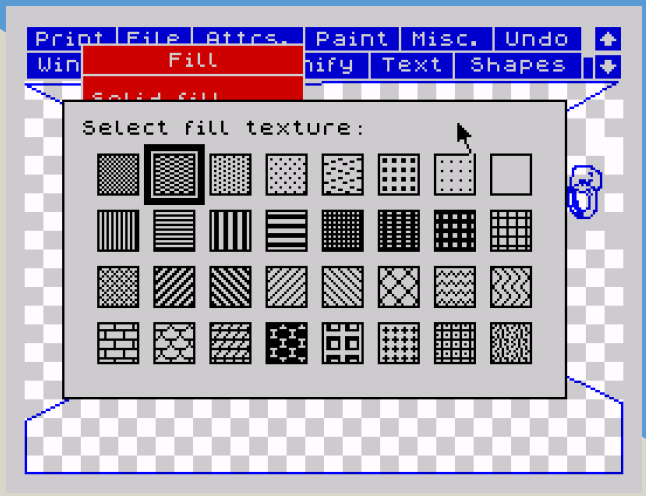
The Magnify menu allows you to magnify an area of the screen to work in detail. This gives you sliders too, so you can move around, as well as a smaller menu set to work with.

The Text menu allows you to add text. The Shapes menu allows you to create various shapes or use line tools.

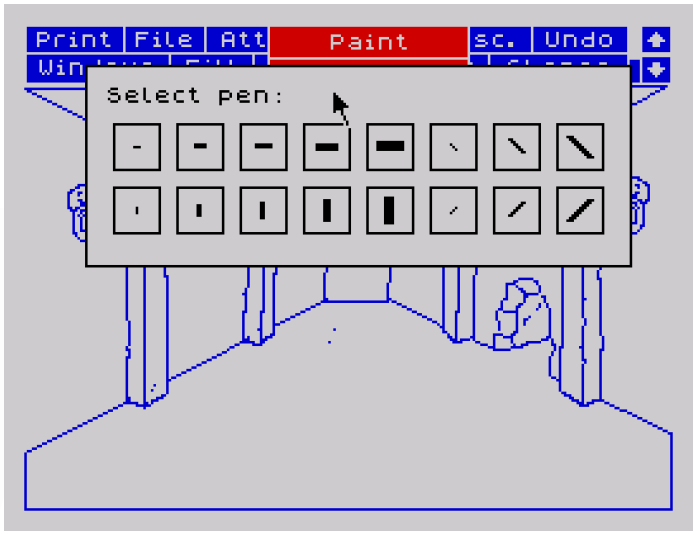
So, a lot of tools at your disposal, but this package won't make you a good artist, you have to have some talent to make anything half decent, as we shall see.

Using all of the tools above, and controlling everything with a mouse means you can quickly start to build up the outlines of your masterpiece. For my poor attempt I wanted to make a dungeon picture. Drawing the lines was easy, and I could fill the different areas with different densities of pattern. This looked really good.

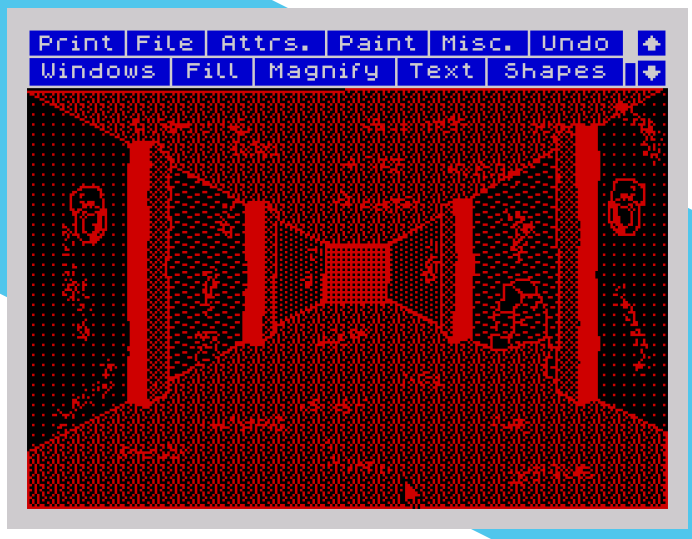




OCP ART STUDIO



Choose your weapon



Now I know this isn't the best picture and I know there is no colour, but I don't have a lot of time to start worrying about attributes, but the tools are all here should you want to get that deep into it. I could use the grid function to check the attributes before I draw, or even to modify my line work to ensure there is no colour clash.

Now lets have a quick tour of the paint options and there are several pen types to choose from. The Spray can gives different densities to use and considering the resolution of the Spectrum, these work well.

The brushes again have a set options plus you can also edit or create your own.

I used the Window tool to grab a metal ring I had drawn on the righthand wall, mirror it and paste a copy onto the lefthand wall. This saved so much time having to try to draw another matching ring.

We have seen the fill tool in action, but you can also create your own fill patterns if you want, which is an amazing addition.

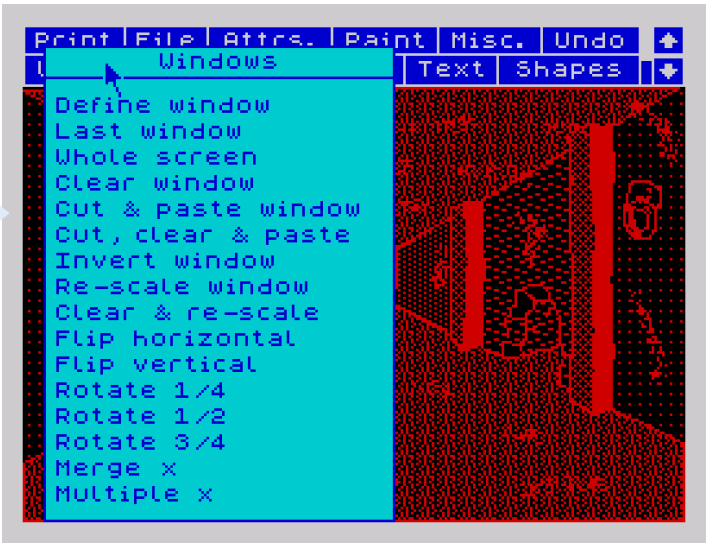
The text options allow normal, double or triple size letters and you can edit your own fonts so plenty of options there. You can even load in other fonts to use.

All in all, this is a very powerful tool, especially when used with a mouse, which is the best option in my opinion. Today's emulators allow you to use your PC mouse as a Kempston or AMX mouse, so you can try this out for yourself.

For any artist, this was a must have for the Spectrum and is highly recommended if you are graphically inclined.

Windows

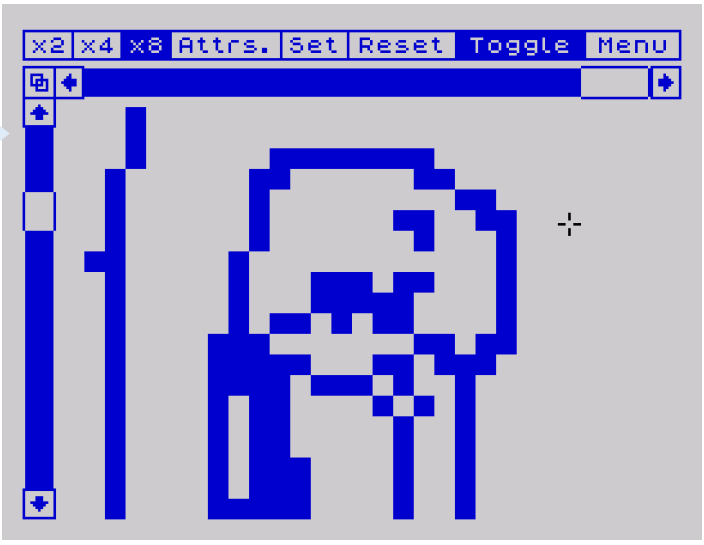
The windows menu lets you cut and paste areas of the screen. This powerful tool is easy to use and helps cut down the time it takes to do simple tasks. Very impressive for a Spectrum.



Magnify

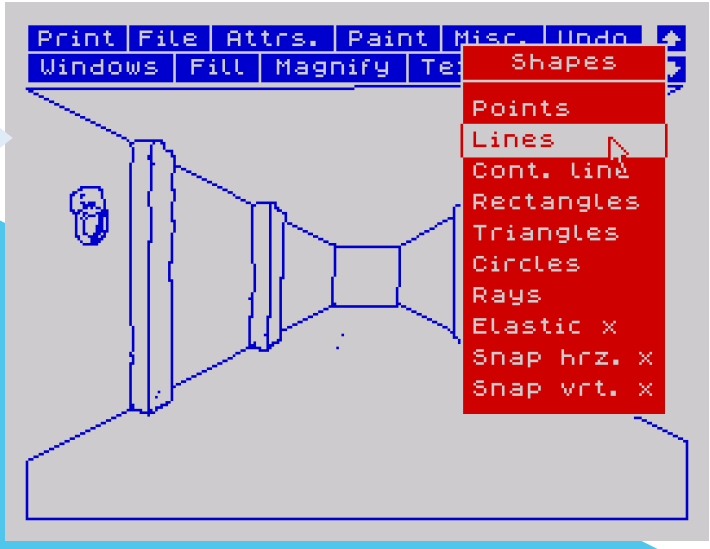
2,4 or 8 times magnification for that detailed work.

Notice the scroll bars on the left and top. These allow you to move around while still in magnify mode.



Shapes

A nice selection of lines, circle and shape options. Easy to use and it allowed me to quickly draw this poor 3D dungeon picture. I added detail to the columns in magnify mode.



Codemasters 1987

In this release from Codemasters, you play a new recruit of the Space Geology College, and being new you have gone and missed the shuttle back to base. Now you have to get back across the deadly landscape with just a jetpac and laser to help you. Could this be an attempt to 'borrow' the jetpac idea from Ultimate's classic? If it was, then it failed.

The game is a cross between Lunar Jetman and Scramble, and not in a good way. There are so many things just not right here.

You fly or walk across the scrolling landscape, able to speed up or slow down, and of course thrust and fire. Walk into anything other than flat land and you explode for some reason. No stumbling, you just explode!

You have a limited amount of fuel, and so have to keep picking up refills, and the same goes for your ammo too. You can shoot the aliens, but it is far easier to just avoid them if possible.

There are teleports that take you further along the level, and these can be very helpful in getting you to the end quicker.

The graphics are large and chunky, and borrow a lot from Jetpac... but sadly does not take the control system.

This game has a kind of fake gravity that means you bounce when you hit the ground and sometimes this sends you into aliens above, very frustrating!

The thrust seems to get more powerful the longer you hold the key, so this again makes control very tricky.

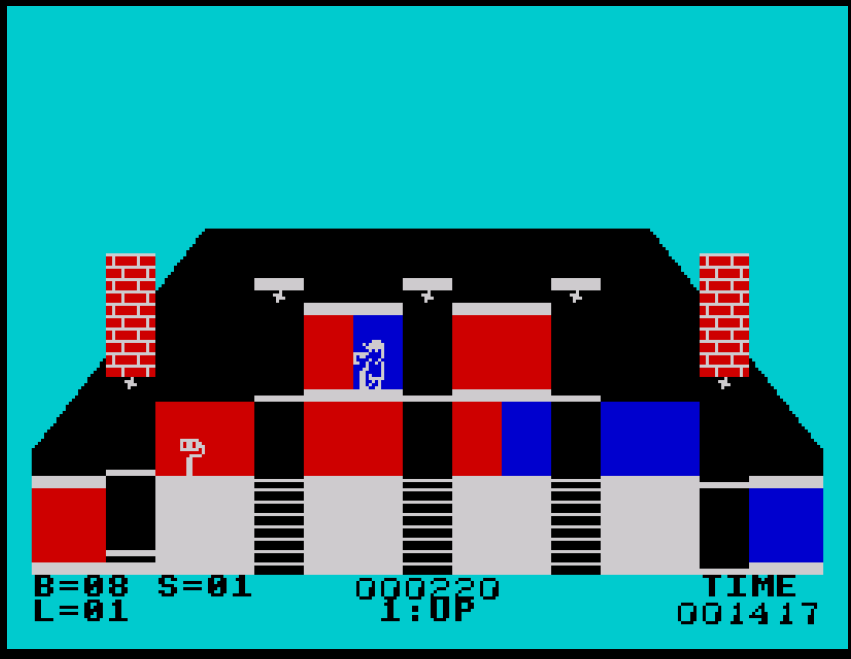
Sound is used well, with a nice tune on the title page and a few spot effects here and there.

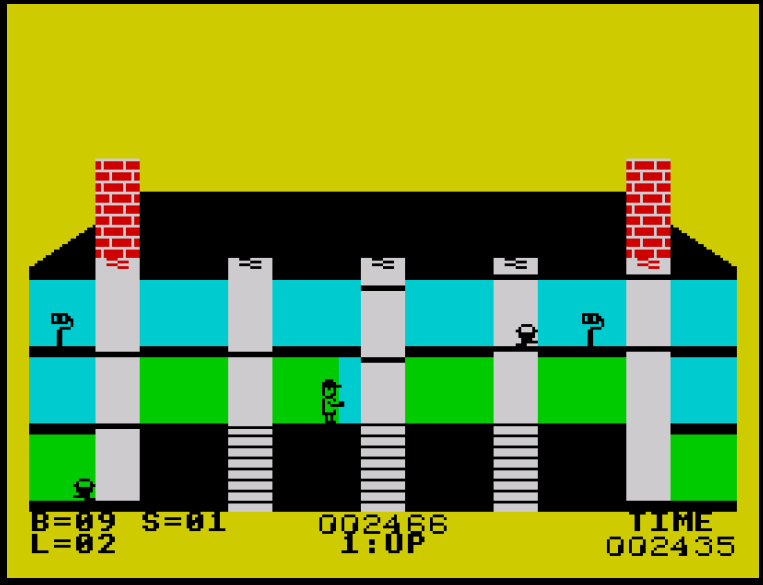
As you progress you get caves and missiles that fire upward, similar to Scramble, and manoeuvring becomes ever harder, sometimes to the point that it makes the game unplayable.

For a budget release this isn't bad if you can get to grips with the control system, but I would have been very annoyed had I bought this at full price.

Bristles

State Soft 1984





At first I thought this game would be a standard Painter or Amidar style game, and it is in a way, but it has some differences. The basic premise is the same in that you have to fill in areas of the screen, but other than that, the game differs.

You control a man who has to paint a house. Each house has a set number of walls to paint in different rooms and on different floors. There are lifts and ladders between floors and you can climb ladders in the normal way. The lifts however work on their own so you have to time your movements to make sure you get in and out at the right moment.

The graphics move well enough and control is crisp, but I wouldn't call them outrageous as the advert suggests. The house is drawn in a way that it's recognisable and the man is a bit on the small side.

There is a time limit too, so you can't hang around, and this makes the timings even more important to getting a good score.

There is a very annoying sound playing throughout the game, a kind of gurgling sound that does not represent anything at all, and this can get tedious.

The game play is not bad actually. The first level is fairly easy but then on the second there are things that can kill you. These knock you back to the start point and you lose a life. Luckily you don't lose any painting you have done.

It's a fun game that will give you a challenge, things move fast though, sometimes too fast, and you can fall down the lift shafts.

Give this one a go.. if you want a nice challenge…

MIND YOUR LANGUAGE

George Beckett continues his voyage through Spectrum programming languages

Optimising the Rabbit Run Game

As we saw in the previous article, a good way to introduce machine code into your programs is to replace BASIC routines that are either not fast enough or that take too much memory. While the program to calculate the day of the week from a date is useful for building up experience, most people who program the Spectrum at some point turn their attention to writing a game. Many Spectrum owners in the 1980's dreamed of writing the next Manic Miner or Horace Goes Skiing and, for this kind of project, some machine code is required.

I was never very good at writing games (and still am not now). However, as for many others, writing Spectrum programs was a bit like building a Lego model. Most of the fun was had during the creation: playing/ using the end-result being less important.

The Rabbit Run program that I introduced earlier in the series is a good example of where machine code can help. The BASIC version of the game runs reasonably smoothly and, with some of the BASIC extensions (like Beta Basic or Laser Basic) we looked at, the performance has been good enough to allow me to include another element in the form of the fox. However, even the fastest implementation—which is probably the compiled, Laser Basic version—suffers from an obvious change of tempo when the fox appears on the screen.

Normally, to look for a bottleneck in a program, you would use a tool called a profiler. However, I am not aware of a profiler for ZX Spectrum BASIC (please let me know if you do), so I typically rely on a mix of intuition, experience and experimentation to identify where a program is running slowly in BASIC. For Rabbit Run, my suspicion is that printing the rabbit (as well as the fox, molehills, and food) is the problem. As mentioned in an earlier article, the Spectrum's PRINT command encapsulates a great deal of versatility: it handles printing to both the screen and printer; it can deal with colour/ attribute modifiers; and can expand tokenised BASIC keywords. However, this versatility means the PRINT command is relatively slow, and using a streamlined machine-code print routine for Rabbit Run could lead to a significant speed-up and reduce the tempo change that occurs when the fox appears.

Normally I would suggest not reinventing the wheel at this point and instead looking for a suitable print routine in a book or online. However, there is valuable experience to be gained by writing your own, streamlined print routine in machine code, as to do so you need to understand how the Spectrum display is laid out in memory; understanding that is crucial for many Spectrum development projects.

As I have hinted, the Spectrum screen layout is not what you might imagine: it's actually quite complicated, though for good reason, as we will see. The display is stored at the beginning of RAM – that is, address 0x4000 (or, in decimal, address 16,384)—and occupies 6,912 bytes. This section of memory is read by a very important piece of electronics inside the Spectrum, called the Uncommitted Logic Array (or ULA, for short), which turns the data in memory into a TV signal.

Recall that the display resolution of the Spectrum is fixed to 256 by 196 pixels, which for printing is treated as 24 rows of 32 character cells, with each character having an 8×8 pixel pattern. Ignoring colour, the pixel pattern for a character can be stored in 8 bytes; by treating each bit as a flag for whether a pixel is ink coloured ('1') or paper coloured ('0'). The pixel information and colour information are stored separately: pixel data is stored in the range 16,384—22,527 and attributes are stored in the range 22,528—23,295.

For the pixel information, the screen is divided horizontally into three sections of 8×32 character cells: for character rows 0 to 7, 8 to 15, and 9 to 23. Within each section, the pixel bit pattern for the first row of each character cell is stored first, character row by character row for the eight rows, followed by the second (pixel) row, then the third, and so on.

So, the first byte of the display file contains the pixel pattern of the first row of eight pixels, for the character at the top-left hand corner—that is, at character cell (0, 0). The second byte of the display file contains the pattern for the first row of character cell (0, 1), and so on until the 32nd byte, which stores the value of the first row of the character at (0, 31). So far, this seems reasonable. However, the next byte (that is, the 33rd byte of the display file) contains the pixel pattern for the first row of the first character on the second row—that is, at location (1, 0); whereas you might have expected to find the second row of the character at location (0, 0). This layout continues until the 256th byte, which stores the pixel pattern of the first row of the character at location (7, 31) and which is the bottom, right character cell in the top, one-third of the screen. The next byte (the 257th) contains the pixel data for the second row of the character at location (0,0) and the subsequent entries up to the 512th byte contain the pixel data for the second row of each character location, as was the case for the first row. Then the third row of pixel data for each character is stored, and so until all eight rows of each character in the top third of the display have been accounted for. This takes 8×256=2,048 bytes. Then the arrangement is repeated for the second third of the screen, and finally for the bottom third of the screen.

You can actually see this layout when you use LOAD "<filename>" SCREEN$ as, for example, at the start of a game. Above you can see the in-progress loading screen of the game Chequered Flag, roughly halfway through loading the pixel data for the second third of the screen. Notice that the top third of the image is complete, but that only some of the pixel rows in the second third of the screen have loaded. Notice also that there is no colour information: this is stored at the end of the display buffer after the pixel data.

The layout of the Spectrum screen is actually quite clever, and designed to make printing text very efficient. If you think a little about the layout, you may notice that each row of pixels in a character cell is separated by 256 bytes (that is, 8 rows of 32 characters). This means that if the address of the first pixel row of a character cell is stored in the HL register pair, then to advance to the second row of the character cell, we merely need to add 256 to the address, which can be done very efficiently on the Z80 with the INC H instruction. If instead the pixel data were laid out sequentially in rows, you would use a sequence such as LD BC, 0x0020; ADD HL, BC, which would take quite a bit longer to execute (and would corrupt the BC register pair).

Following on from the pixel data is the attributed data, which starts at address 22,528. The attributes for each eight-by-eight character cell are encoded as a single byte with bits 0—2 representing the ink colour; bits 3—5 holding the paper colour; bit 6 holding the brightness; and bit 7 holding the flash value. For example, an attribute value of 75 (which is 01001011 in binary) represents magenta ink on blue paper with bright enabled and flash disabled.

The attribute data is laid out in the obvious manner, one row at a time in sequence. So, the attribute data at print location (X, Y) is stored at address 22,528+32*X+Y.

If the above explanation seems confusing, then you might want to try experimenting with the display, by writing a BASIC program to print a character (perhaps a user-defined graphic) by POKE-ing the pixel pattern into display memory, something like Fig 1.

As written, the program will display the first user-defined graphic at character cell 0, 0. Can you modify the program, so that instead it prints the character to another cell location? Can you POKE a suitable value into the attribute space to change its colour? Can you make the program print the user-defined graphic upside down? So long as the address used in the POKE command, in line 30, is always between 16,384 and 23,295 (that is, in the display area) you will not cause any harm: at worst, you can enter CLS to reset the display. However, if you inadvertently POKE numbers higher up in memory, you might crash the Spectrum and lose your program.

Armed with an understanding of how the Spectrum display is laid out in memory, we can now write an efficient print routine, to display a user-defined graphic anywhere on the display, such as in Figure 2.

Fig.1

```
10 LET p=16384 : REM Location in display to print to
20 FOR n=0 TO 7
30 POKE p+n*256,PEEK (USR "a"+n) : REM Advance by 256 bytes for each pixel row
40 NEXT n
50 STOP
```

Fig.2

```
DISPLAY:      equ 0x4000      ; Start of display buffer
ATTRIB:       equ 0x5800      ;Start of attribute buffer
UDG:          equ 0x5c7b      ; System variable holding address of UDGs

      org 0xfe00              ; Address is 65024d

      ;; On entry:
      ;;   A contains UDG character to be printed (upper case),
      ;;   DE contains row and column of print location;
      ;;   B is non-zero if there is attribute data; and (if so)
      ;;   C contains the attribute data
PRINT_UDG:
      push bc             ; Save the attribute value
      sub a, "A"          ; A contains character count from first UDG
      add a,a             ; Multiply by 8, as each UDG occupies 8 bytes
      add a,a
      add a,a
      ld b, 0x00                ; Load offset into BC
      ld c,a
      ld hl,(UDG)               ; HL is base address of UDGs
      add hl, bc                ; Apply offset so HL now points to pixel
                                ; pattern for first row of UDG

PRINT_IT:
      push hl                   ; Save character address
      ld hl, DISPLAY            ; HL points to start of display memory
      ld a,d                    ; Row to print to
      cp 0x08                   ; Check if in first screen region
      jr c, ROW_CALC            ; Jump forward, if so
      ld hl, DISPLAY+0x0800     ; HL points to second screen region
      cp 0x10                   ; Check if in second screen region
      jr c, ROW_CALC            ; Jump forward, if so
      ld hl, DISPLAY+0x1000     ; HL points to third screen region

ROW_CALC:
      and 0x07                  ; Only need lowest three bits of row number
      rrca                      ; Move offset into upper nibble and multiply
      rrca                      ; by 2
      rrca                      ; Upper three binary digits
      or e                      ; Apply column offset (lowest 5 binary digits)
      ld c,a                    ; Move offset into BC
      ld b,0
      add hl, bc                ; HL contains offset to print position

      ex de,hl                  ; Save AT coordinates to stack, swapping with
      ex (sp),hl                ; character address (from stack)
      ex de,hl
```

```
       ld b, 0x08           ; Eight rows of pixels
PLOOP: ld a, (de)           ; Next pixel pattern to print
       ld (hl), a           ; Display
       inc de               ; Advance to next character row
       inc h                ; Advance to next display row
       djnz PLOOP

       pop de               ; Restore AT coordinates from stack

ATTRIB_CHECK:
       pop bc               ; Restore attributes
       ld a,b
       and a

       ret z                ; Exit if no attributes to update

COLOUR_IT:
       ld h,0               ; Load row into hl
       ld l,d
       add hl, hl           ; and multiply by 32
       add hl, hl
       add hl, hl
       add hl, hl
       add hl, hl

       ld a,l               ; Apply column offset
       or e
       ld l,a

       ld de, ATTRIB        ; Finally add offset to start of
       add hl, de           ; display buffer

       ld (hl),c            ; And update attributes

       ret
END:
```

As you may notice, the downside of the Spectrum screen layout is that it is an involved process to work out the address in the display buffer that corresponds to a particular print position. However, for most printing operations, this is computed rarely, making it a worthwhile trade-off to have fast character printing.

Based on a crude timing experiment, the above routine is around four time faster than the BASIC equivalent. I therefore tried replacing the BASIC print commands in the Rabbit Run game with calls to this routine—via a suitable wrapper code, something like:

```
PRINT_CHAR_WRAPPER:
Ld a, 'A'
Ld BC, 0x0000      ; Attribute data
Ld DE, 0x0000      ; Print coordinates
call PRINT_UDG     ; Call custom print routine
ret                ; Return to BASIC
```

—into which the UDG, attribute value, and coordinates are POKE-ed before each USR call.

Unfortunately, the version of Rabbit Run that uses this optimised print routine is not noticeably faster than the original BASIC version, which is disappointing and is probably because of the overhead of having to POKE up to five values into memory before each USR call.

Undaunted by this setback, I looked at ways to port more of the time-critical elements of the game to machine code and quickly came to the conclusion it would be worthwhile to re-write the main game loop in its entirety in machine code: having written this print routine and having some integer arithmetic routines to hand from the machine-code version of Zeller's Congruence (see previous article), it became primarily an integration activity.

Looking through the BASIC listing of Rabbit Run, I noted a number of functions for which I needed a machine code version:

- To generate random numbers – for example, to position new food and molehills;
- To produce game sounds;
- To read the keyboard, so that the player could move the rabbit.

There are various random-number generators available from the Z80 Heaven website. I selected one that produced 16-bit random numbers, as this was flexible enough to cover the different uses in Rabbit Run. In particular, by treating a 16-bit random number as two 8-bit coordinates and then using the DIV8 routine from Zeller's Congruence to work out the remainder after dividing the coordinates by 24 and 32, respectively, I can quickly create random coordinates for new molehills and food.

Making Noise

To produce game sounds, I used a ZX Spectrum ROM routine, called BEEP (at address 0x03b5 and described on page 11 of Logan and O'Hara's "The Complete Spectrum ROM Disassembly"), which takes two inputs similar (though not quite the same) to the BASIC BEEP command. I wrote a short wrapper function for this, to play a sequence of sounds—for example, for when the rabbit dies - based on some pre-computed parameters, as shown if Fig 3.

Input

The ZX Spectrum ROM has several different routines for reading the keyboard and these are described in detail in Toni Baker's book "Mastering Machine Code on Your ZX Spectrum". However, none of them does quite what I wanted and as I only needed to check four keys, I decided to write my own routine. The ZX Spectrum keyboard is divided into eight sections and the status of the keys (that is, pressed or not) in each section is determined by reading the appropriate port with IN and testing the value of the corresponding bit. For example, the 'Q' key is tested by looking at the value of bit 0 of port 0xfefb. It will be reset if the key is being pressed, and set otherwise. The code snippet in Fig 4 tests if the 'Q' key is pressed:

Having written or integrated these different functions, I was able to write the main game loop and test it.

Following a bit of debugging in Spectaculator, I got a working version of Rabbit Run. However, it ran far too quickly to be playable. The rabbit moved three or four squares whenever I pressed a direction key, and the fox raced across the screen to catch it in around a second.

Fig.3

```
;; One entry HL points to the start of a list of parameter pairs for BEEP,
;; terminated by 0x0000. This routine corrupts most registers.
BEEPER:
ld e,(hl)           ; Retrieve first BEEP parameter
inc hl
ld d,(hl)
inc hl

ld a,d              ; Check for zero, which indicates end of
or e                ; list and return if so
ret z

ld c,(hl)           ; Retrieve second parameter
inc hl
ld b,(hl)
inc hl

push bc             ; Move BC into HL and save list pointer
ex (sp),hl

call BEEP           ; Call ROM routine

pop hl              ; Restore list pointer

jr BEEPER
```

Fig.4

```
;; Check for up - 'Q'
UP:    ld a, 0xfb          ; Read status of port 0xfefb
       in a, (0xfe)

       bit 0,a             ; Check if 'Q' is being pressed

       jr z, MOVE_UP       ; Move rabbit up, if it is
...
```

This was a good outcome, as the reason for porting the game to machine code was to improve the speed: the next step was therefore to refine the timing of the game, which I did in three stages.

First, I added a HALT statement at the start of the game loop. The HALT command tells the Z80 to wait for a system interrupt, which is issued fifty times per second on the Spectrum at the beginning of a screen re-draw. As well as producing predictable timing, putting HALT at the start of the game loop has the second advantage of making sure the graphics are drawn at the beginning of the screen re-draw—before the scanline gets too low down the screen, which would cause flickering.

With the HALT statement inserted, I have a timing reference for the game loop, which cycles fifty times per second, whether or not the fox was in play—so no slow-down. As I only want the fox to be able to move about 6 cells per second, I introduce a conditional check that only lets the fox move every eight game loop iteration.

Finally, it was not easy to press a direction key for just one fiftieth of a second, so the rabbit tended to move multiple squares when a key was pressed, which made for poor game play. I therefore introduced a four-cycle delay whenever the rabbit moved before the keyboard would be checked again. This provided a good balance with the player both able to quickly move the rabbit and precisely navigate between molehills.

With these timing changes implemented, the game has the features, difficulty and dynamism of the original BASIC version, but without the timing problems and slowdown. The machine-code version of the game also occupies just under 1 kilobyte of memory (not including the few Spectrum ROM routines that I used).

You can download the source for the game from The Spectrum Show website, along with a TZX image to play in an emulator. There are lots of ways in which this game could be improved, so why not give it a go. It will help you validate your growing machine-code skills and, once you have something you are happy with, you can share it with your friends—just like people used to do in the 1980's!

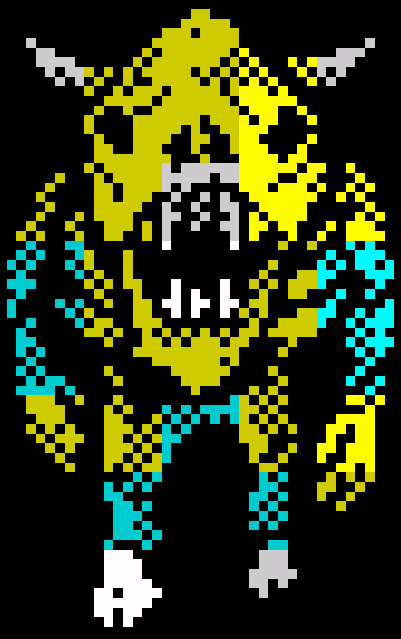
In the next issue we will be looking at some of the other programming languages that can be used on the Spectrum, focusing on a language called FORTH, which fits somewhere between the ease of BASIC and the performance and compactness of machine code.

Downloads from this series can be found at www.thespectrumshow.co.uk/basic

More from George next issue.

the DARK

Oleg Origin 2016



The Dark, originally released in 1997 by Oleg Origin, was re-released in 2016, and what a game this is. From the very first load, the game impresses with a spoken story accompanied by excellent pictures and an understandable voice - even on a 48k machine.

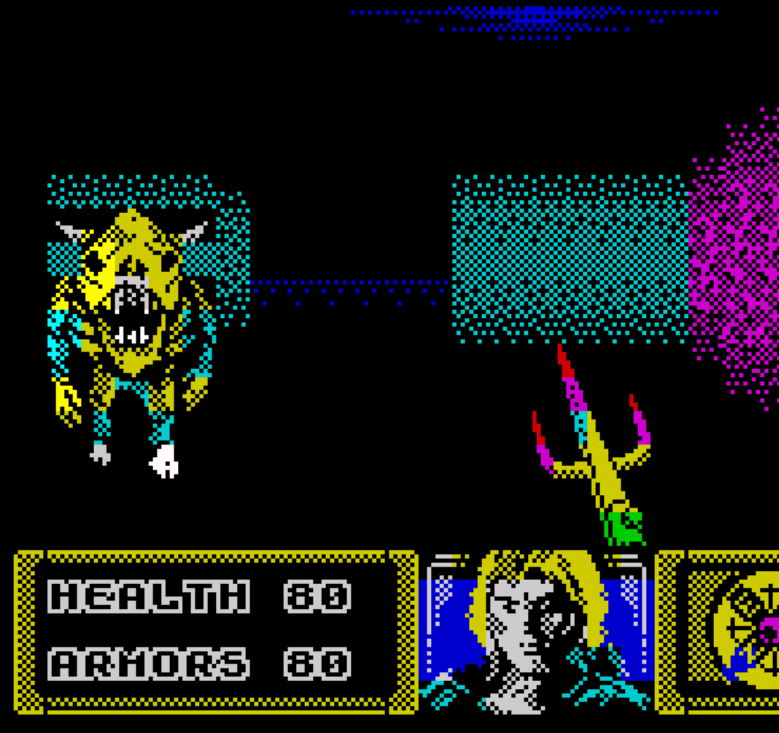
Onto the game then and here we have a very impressive 3D action game that many Spectrum fans have dubbed the Speccy version of Doom. The game is not actually like doom, but the comparison comes from the graphics style and gameplay. Yes, a full screen first person shooter on the Spectrum, I can hardly believe my eyes.

You travel through the 3D world looking for the exit and these usually need a key to operate. Keys are represented as hand prints in the walls, but your first task is to get a better weapon than a pitch fork.

You are not alone in this world though and there are other beasts that are not happy with you being there. They wander about and when you get in range, they charge towards you with deadly intent.

A pitch fork is not the best weapon and soon you should find a hand gun. This makes dispatching the enemies much easier as long as you don't run out of ammo.

You still have to keep out of there way, so it's dodge and attack strategies. Not as easy as it seems, and I didn't get very far on my first few attempts, or in fact my attempts after that too until I located the gun.



You can view a map and this indicates keys and exits, as well

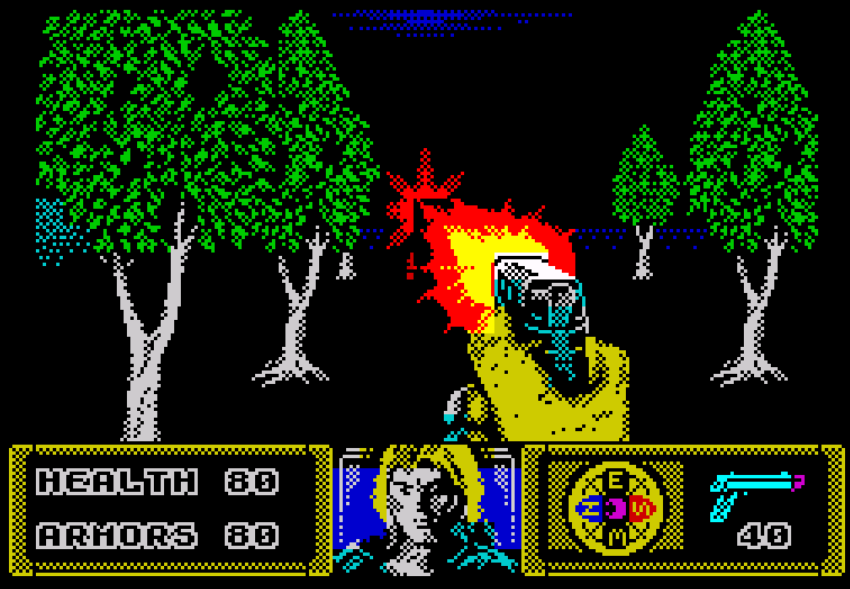
as trees and other items. This is essential if you want to complete a level. The game pauses when viewing the map, so you can take a breather here as well.

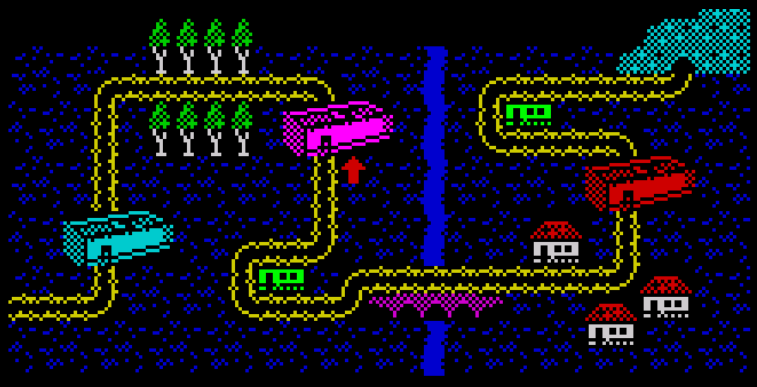
Back in the game and the graphics are great. Very colourful, and although there is not the full 3D walls like Doom, the effect is excellent. You hardly get a chance to view the scenery due to the enemies. The scenery consists of walls, trees and other objects, all coming together to give a great effect.

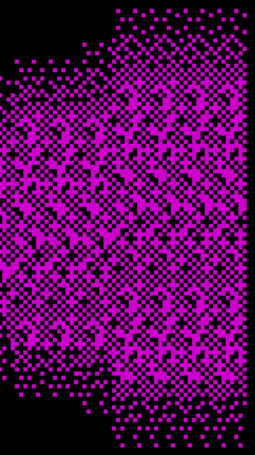
Sound is used well with background effects and footsteps as well as firing and death sounds. Control is by keyboard or joystick and is very responsive. That's just as well really considering the amount of enemies around.

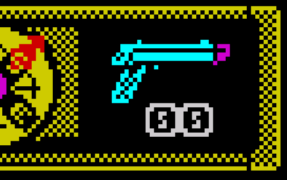
This is not an easy game, at least for me, but it is well worth a play just to see the Speccy playing a 3D shooter.

Definitely one to play and a brilliant achievement on the humble Sinclair machine.

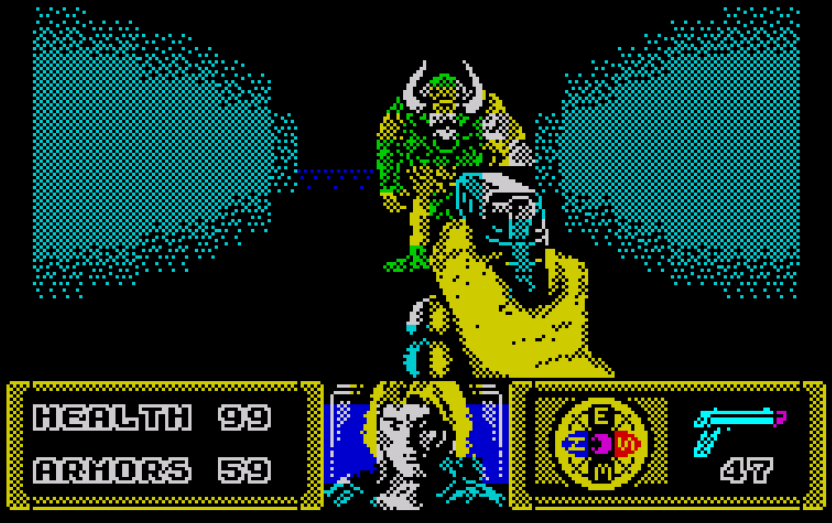






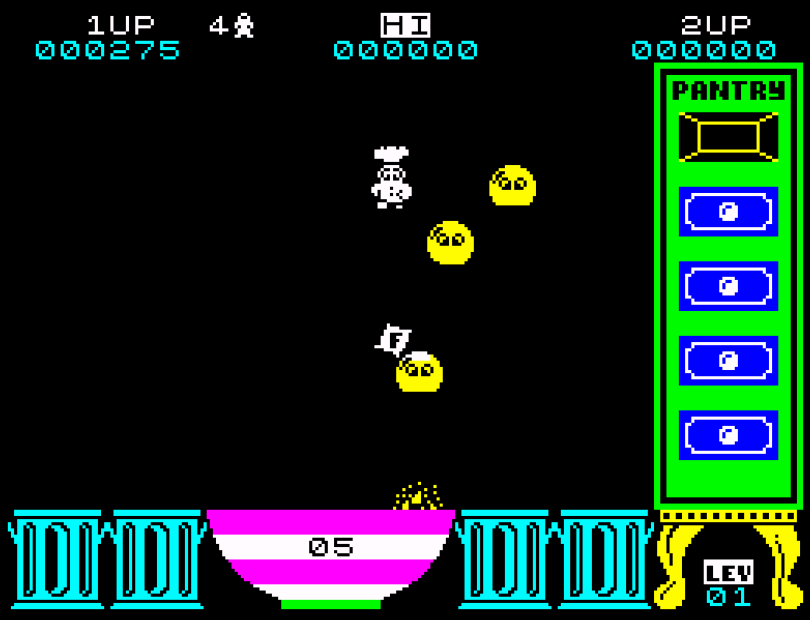


MONSTER APPROACHING!



Cookie

Ultimate Play The Game 1983



Is there anyone who doesn't know Cookie?

In 1984 Ultimate Play The Game released four games onto the market to announce their arrival and pushed game quality out of the dark ages. Cookie was one of these 16k titles and was also released on ROM cartridge.

The idea is simple and yet expertly executed, it was clear the company had arcade links, and the other three games were also as playable.

You control a chef trying to bake a pie - yes that simple. To do this you have to herd the ingredients into the bowl at the bottom of the screen using just your bags of flour. Each set of ingredients appear in turn from the cupboard on the right and you have to force them into the bowl. Ingredients can only get in the bowl if they have been hit by flower, so you can't just let them float in on their own.

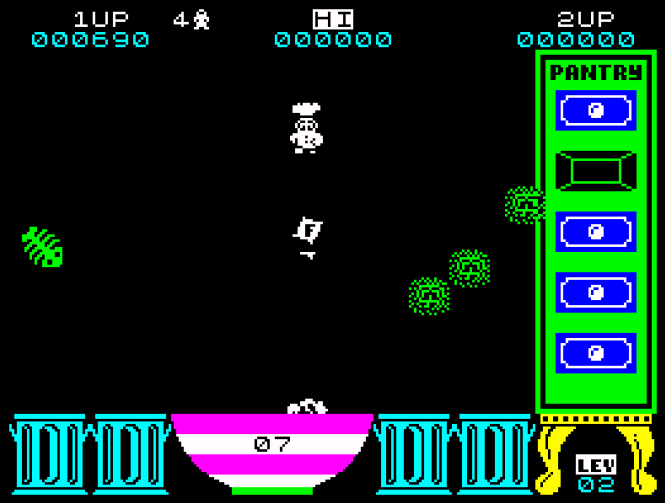
Each ingredient is progressively harder to beat and there are other things to avoid too, like the bin monsters throwing bits of garbage at you. If any of the garage gets into the bowl then the ingredient count goes up and you need ten items before you can move on.

A count of the required ingredients is displayed on the bowl and once they are all in, it's time for the next lot. If you manage to get all five ingredients into the bowl, the pie is baked and the game begins again.

The graphics are top notch. Smooth movement and well defined, easy to control and they really suit the game. Sound is also good with occasional music and nice effects throughout.

I like the early ultimate games, they have a certain charm to them, and they are certainly games you keep going back to.

A golden oldie then, that's definitely worth a quick play.



STARCLASH

Derek Brewster 1983

This early shoot-em-up was written by Derek Brewster, a well known programmer who later went on to the *Code Named Mat* games as well as *Kentilla* and *Curse of Sherwood*.

This though is a version of the arcade game Astro Fighter, a simple, yet tricky little shooter from 1980.

Waves of aliens move down the screen and to progress to the next level you have to clear all aliens before they reach the bottom. There is no second chance, should they reach the bottom, they all respawn again.

If you are successful, the next wave begins with different aliens and different movement patterns. As the aliens get less, they move faster, making things even more difficult, not to mention they fire back at you and of course there are meteors to worry about too.

The different aliens are shown at the top of the screen, with the current wave flashing. The attack waves move in the same way as the arcade machine, making them tricky to hit.

The graphics are quite basic but move smoothly and the sound is very good for a 16K game. There is a constant warble sound along with firing and explosions.

Control is good but the game does slow down when there is a lot on screen.

For a 16K shooter, this is a nice little pick up and play game. Easy to get into, easy to control and quite addictive.

LOCK AND LOAD

SOFTWARE PROTECTION THROUGH THE YEARS

Wherever there has been something that can be copied, especially software, there has inevitably been copy protection schemes. They have been in place since the early days of Spectrum gaming and were created to prevent the one obstacle that was hindering many companies: piracy.

Overtime, they became more and more complicated, but occasionally, the schemes themselves would get in the way of legitimate users, anyone remember Daley Thompson's Decathlon?

Protection began in a very simple way, disabling the break key. This method was relatively simple at first and involved hacking the system so that if BREAK was pressed, the system would crash.

Examples of this kind of protection include Horace Goes Skiing and Incentive's Millionaire. It was however incredibly easy to bypass and you could just issue a MERGE command to see the BASIC listing before the loader started.

Most copying software of the time would be able to copy this type of program. Since the data was saved with the ROM routines, it was also incredibly easy to just make direct tape to tape copies.

Another sneaky method was to try and fool the user in thinking the game was loading machine code when it was in fact loading BASIC. A good example of this is Transylvanian Tower by Richard Shepherd Software. The game begins loading with the BYTES prefix, but is actually BASIC. You can break into the game code by holding down a certain key combination.

Losing Your Head

The next stage was headerless data blocks.

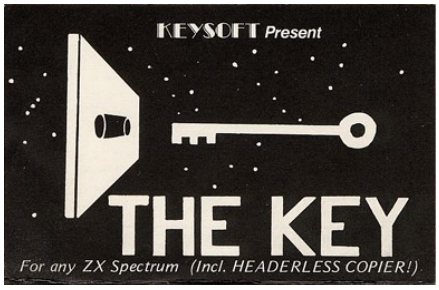
A slightly more advanced method from around the same time modified the data so that it was impossible to load the code direct from BASIC, instead requiring a special piece of machine code.

Usually code was loaded in two section, the header – which contained details about size and location, followed by the data block itself. Without a header though, straight forward copying was difficult.

This kind of loading can be seen in 3D Deathchase.

While this protection scheme was difficult to break for casual users, peo-

ple with a basic knowledge of machine code found it easy to crack. Most often, you could just use the MERGE command to prevent the game from running straight away after loading. From there, you could inspect the BASIC listing and try to figure out how the game was loaded. The Key (a software copying program) could also copy most software that used this technique.



Another early trick, only suitable for programs written in BASIC, involved putting invalid control codes inside the listing. While the listing itself would run correctly, it would stop listing when the invalid control code was reached.

Other devious tricks relied on the way Sinclair BASIC worked. Numbers were stored twice: once in the floating point format used for calculations, and once in ASCII format for printing. Clever programmers changed the ASCII values but kept the floating point numbers. The computer would execute the correct command since it relied on the floating point number, but the listing used the ASCII number, so the incorrect value was displayed on the screen.

Defeating this protection would require the hacker to list the proper values using the PEEK command, something that would require a good knowledge of BASIC addresses to achieve.

Things began to get a little more technical as the industry evolved and special loaders were used instead of the standard Sinclair ROM routines.

Land of the Loaders

Speedlock

Speed loaders were created to load files faster than the original ROM routines and act as protection at the same time. The most infamous of these was Speedlock which was programmed in 1983 by David Aubrey-Jones and David Looker. Ultimate Play The Game, US Gold and Ocean were among those who used Speedlock extensively for their programs.

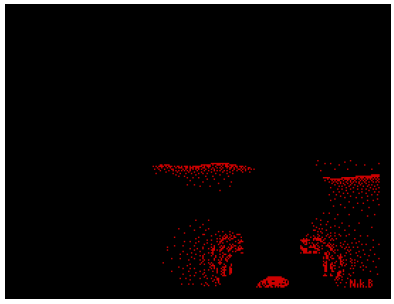
This loading scheme had a distinct sound with a set of audible 'clicking' tones in the leader tone of the data. This, and the high speed, made Speedlock very difficult to copy using tape to tape. However, copiers for Speedlocked programs soon cropped up, such as the Lerm tape utility.

When originally released this also caused many players to complain that their games simply would not load. This was usually down to miss-aligned tape heads, but with such a high degree of accuracy needed, it just annoyed the paying public.

Alkatraz

Alkatraz was another popular protection scheme and had the ability to animate loading screens during the load process. Examples of this loader can be seen in Cobra, Bobby Bearing (where the game's plot is explained in a scrolling message) and Trantor The Last Storm Trooper.

Like Speedlock, it was difficult to copy using tape to tape, but also like Speedlock, copier programs soon cropped up for it.



Bleepload

Bleepload was a strange loading scheme, because it loaded programs in blocks of 250 bytes each. The amount of blocks and the gap between each block made Bleepload programs very difficult to copy using a standard copy program. Examples of this are Gunstar and The Pawn, as well as many firebird games.

This block loading system also had a serious disadvantage: the separate blocks meant that most of the tape was taken up by leader tones, which slowed down loading considerably.

As the schemes became more and more difficult to crack using software programs, it was time for another approach – hardware.

Bring On The Hardware

All schemes were null and void when the Multiface arrived from Romantic Robot. It could copy all programs released just by pressing a single button on the device. Once halted, the user could then save all of the RAM to tape. The built in monitor also meant that all games could be inspected and hacked. This was a great problem for the software companies, as it allowed copying of any program to occur simply through the press of a button. The only game known to defeat Multiface copying is Super Wonder Boy by SEGA, but this was mainly because the title was multiload and therefore unsuitable for normal Multiface copying.

With software and now hardware schemes being defeated, the companies turned to yet another option, off-tape protection.

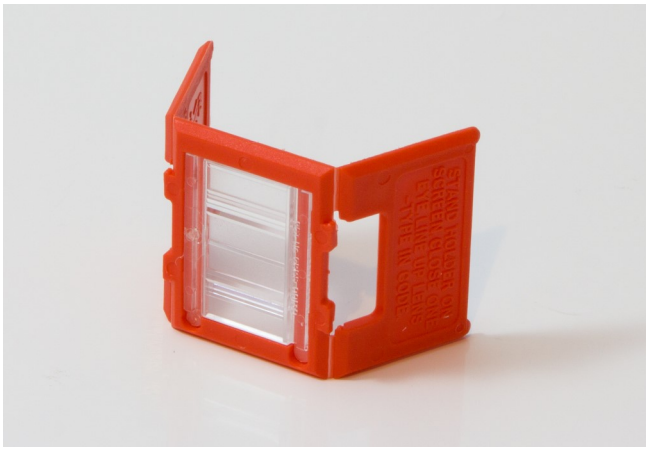
Reading Aloud

The most notable protection scheme was the Padlock scheme used by Software Projects. This involved a coloured card which contained codes for the game. When the game was loaded, it asked for a code on the card. If the code was wrong, the player was given another chance to enter the correct code before the computer reset itself.

The tape itself was easy to copy, but the coloured card was difficult to reproduce at the time since colour photocopiers were very expensive and certainly not within the reach of the average user. However, the tape itself was sometimes unprotected (such as with Jet Set Willy), so it was easy to go in and POKE the code after the game had loaded. Software Projects would not offer replacement cards, so if the card was lost, the game was unplayable.

Users also circumvented this by hand writing out all of the codes and colours, and this could then be easily photocopied.

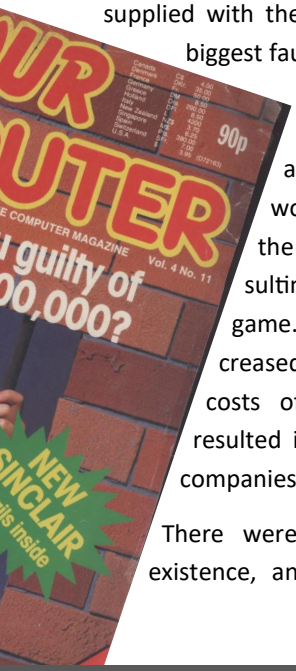
Most Annoying Lump Of Plastic Ever...



Lenslock!

This was perhaps the silliest type of protection scheme. It used a physical device which consisted of a bit of plastic inset with a set of prisms. A scrambled code would appear on the display and the Lenslok would be used to descramble the code.

Lenslok would not work on televisions that were too large or too small and the adjustment instructions originally supplied with them was unclear. The biggest fault was that occasionally, the Lensloks would be mixed up and the wrong one would be shipped with the wrong tape, resulting in an unplayable game. Lenslok also increased the manufacturing costs of the game, which resulted in less profit for the companies.



There were other schemes in existence, and different versions of the ones mentioned here.

Some relied on words on the inlay or in the manual of the game. These particular types were easily circumvented, and like all protection schemes that came before, proved that they only stopped the playground copiers, and had very little impact on mass piracy.

Today, if someone wanted to copy your game, they could and there was very little you could do to stop it.

Top Secret

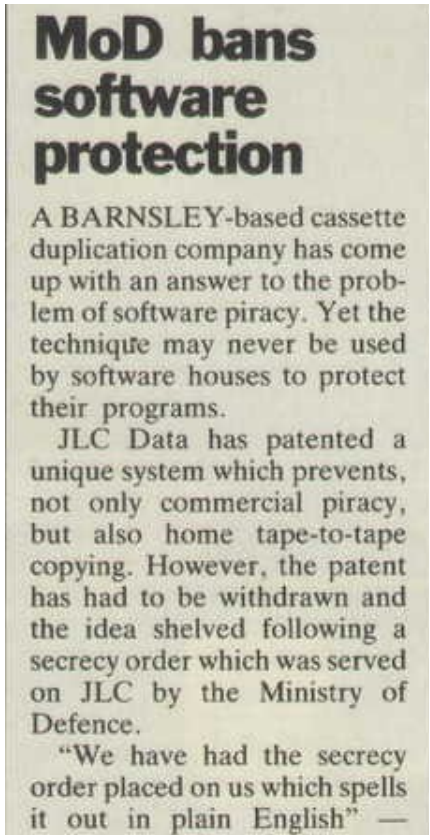
Modern games now come with internet protection or require always on connection to the company's servers – which wasn't available during the Spectrums life. This made it necessary to write, modify and invent software protection.

A special mention must also go to the device created by JLC that was blocked by the Ministry of Defence because it was too good and could be used to protect data they may want to get at.

From what is known, it consisted of a small chip inside the cassette tape that had to be there for the software to run. A second signal is then imprinted underneath the main data on the tape that checks for the presence of the chip.

Because the MOD confiscated all material relating to this, the public will never really know what it actually did.

From the data available though, unless



MoD bans software protection

A BARNSLEY-based cassette duplication company has come up with an answer to the problem of software piracy. Yet the technique may never be used by software houses to protect their programs.

JLC Data has patented a unique system which prevents, not only commercial piracy, but also home tape-to-tape copying. However, the patent has had to be withdrawn and the idea shelved following a secrecy order which was served on JLC by the Ministry of Defence.

"We have had the secrecy order placed on us which spells it out in plain English" —

the device continually checked for the chip, it would be easy to copy once the game had been loaded into memory.

Copy protection will always be around despite the fact it is often only valid users who are affected by it, and it is often overlooked in the modern world of SD card interfaces and emulation that load games instantly. I bet there are few of you who actually sit down and load Spectrum games at normal speed.. and to be honest I don't blame you.. it was interesting to see some of these loaders in action though, especially Alcatraz with it's animation features…

Something often overlooked when emulating older machines.

Reviewing the games that came with the Vega console

..but without instructions!

A.T.A.C.

This game looks great, and luckily there is little need for instructions; it's a horizontal shooter.

You fly a helicopter that looks very similar to Airwolf, and have to fly across a parallax scrolling landscape from right to left. You have to keep the copter in the air by continually pressing the up key, which is a bit of a pain.

Enemy helicopters and planes come at you from both sides, and you have to spin around to take them out before they crash into you.

You have a shield that slowly depletes if any enemy hits you and a fuel limit too, and you must also stay away from the ground or you just explode.

If you manage to complete a stage, there is a bonus round where you shoot holes in a wall and then shoot the moving vehicles - not much fun.

The graphics and scrolling are great, and this accompanied by some nice sound effects make for a very playable game, ideally suited to the little handheld.

The game is not too hard, letting you progress far enough to want to try again, but I could see no way to replenish your shields, at least until many plays later, then I noticed I had been shooting them!

DEMONSLAIR

What a terrible, un-controllable game this is!

You are first asked to enter your name, so this means pulling up the Vega keyboard and entering your initials. This is before you can even get into the game!

The first and major problem is the control system, and unlike most of the games on the Vega, this one has not been setup to use the buttons. This means that to jump, you have to press the UP on the D-pad!

This makes getting anywhere a complete nightmare, and soon puts you off the game. It is very hard to jump and move left or right at the same, so more often than not, you will fail and get hit by an enemy.

You control a wizard who has to collect four letters of a word on each screen. Once all letters have been collected - in the right order, a key appears. Collecting this will then let you move to the next screen via a door.

The graphics are large, colourful and move well, but controlling them is impossible. Trying to time your jumps to miss the enemy sprites is tricky, and many places on the screen are impossible to get to unless you bump into one.

There are a large number of sprites in the game, all drawn well with limited animation, and the backgrounds are nice too.

Sound is a let down too. No sound for walking or jumping, just for collecting a letter and when you bump into nasties. Bump into too many, and you lose a life.

All in all, a bit of a mess, that could have easily been improved by making one of the buttons act as the UP command.

On a side note, when trying this game in an emulator, the Kempston option also seemed broken, and I had to use the keyboard to grab the screen shots!

WARNING: MAY CONTAIN SPOILERS

GRUMPY OGRE'S Adventure Page

It is not often I get the time to immerse my old bones into a large, deep adventure these days, but accidently I find myself in such a position.

I was skimming through some adventure titles looking for something I could play and subsequently write about here, and I happened to stumble on Mordon's Quest from Melbourne House.

Those lovely people at MH gave us the Hobbit and, via Abersoft, Advanture One - renamed to Classic Advanture.

Mordon's Quest however, is a later game and I was intrigued to find out what it was like, and I was very pleasantly surprised. Take the opening text, very atmospheric, and it dragged me in.

```
You awake from a deep, troubled sleep,
to find you have fallen from a large
four-poster bed.....

You are in the master bedroom, despite
the richness of the bed and the oriental
carpet underfoot, the room is sparsely
decorated. An indefinable atmosphere
permeates the room, as if some great
tragedy once occurred here. Tall sash
windows face north and west. Large
double doors open to the south.
```

Present in 40 column text, the descriptions are wonderful and really set the scene, and things begin to unfold as you play.

You start in a house and after a bit of exploring, climbing and wandering about in the mist, a mystical figure appears and tells you he is Mordon. After some gibber gabber, he offers you a quest.. and then the journey really begins.

Revisiting an item in the house will whisk you off to the next section, the Jungle.

The game is not without it's problems though, and some familiar commands do not work. For example, to get a description of a room, you would normally type L or even LOOK, but this game recognises neither. Instead you have to type WHERE AM I to get the location described again.

There is also one of those most hated elements, a maze, and you will have to make a few leaps of faith here before you can move on.

The thing that makes this game though, is the vast amounts of text and detailed descriptions. They remind me of old online text games where space was not an issue and the authors could write masses of details to keep you enthralled.

I know text only games are not everyone's cup of grog, but I really enjoyed playing this game, it was such a refreshing change from the usual YOU ARE IN A CAVE.

The game is large and will keep you challenged for many many hours, if not days or even longer.

For any adventure fan, I would also point you in the direction of a great documentary called Get Lamp.

This covers the very start of adventuring and incorporates interviews with many of the famous people involved. Scott Adams tells the story of Adventure International and there is plenty of Infocom material.

You can buy this from the official website (www.getlamp.com) or watch it on Youtube. I chose to buy it because I wanted to show my appreciation for the tremendous effort that went into this film.

Moving on to another Melbourne House game, Castle of Terror.

This is such a different beast altogether, and one I am not drawn to.

The graphics are nice, but the game is just odd. There are many reported issues and seemingly that game has never been completed.

I first found a problem when I entered the house close to the start. In here is a table, on the table is a knife. Picking up the knife was fine, then after some walking about and getting nowhere, I re-entered the house and re-examined the table. The knife was back there!

I picked it up, and it appeared in my inventory. I examined the table, and the knife mysteriously was back there!

There are also items drawn in the pictures that are key to the game, but are not in the description. For example, stood outside the mill there is a magenta thing next to the steps. This is a barrel that you have to examine. It is not in the description!

After this discovery, I then went about examining everything in every image - but it didn't help me very much.

Even when you get inside the mill, there is a lantern. No mention of this in the description and it isn't even drawn on screen! I had to find this out by taking a cheeky look at the solution text!

That is no way to treat adventurers. All this accomplishes, it a hatred of the game and players soon getting board of having to guess what or where things are.

It was while reading the solution that another sneaky trick emerged - there are two identical locations in the game (the Mill) each containing different objects!

I then ventured into the local pub where I met a man in need of a drink. Luckily I had helped some locals dig a field and they had given me a coin. Haha... BUY ALE then informed my I had no money! Yet another mysteriously vanishing object.

This was all too much, and having spent a good hour getting nowhere, I reset the emulator and went for a tankard of ale myself.

Melbourne House produced a mixture of games, the great and the awful I encountered here!

Oh well... onward to my next pint and some more adventures.

BOOKSHELF

Yes, it's back. He best Spectrum mag returns for a one-off special in 2017. Chris Wilkins, the man behind so many other books about the Speccy, has produced a fantastic effort to bring this classic magazine back.

But hold on, why is a magazine featuring in the book section? Well, it isn't really a magazine, it's a hard back book that just happens to be the same size as the original magazine and features all the content of the original.

It was a real treat to get this just before Xmas, and it truly was special to open the parcel and start reading the thick glossy pages.

The quality of the paper is superb and this in turn produces crisp and colourful images on every page.

So what does it contain?

If you have ever read any of the original Crash magazines, you will know, and it does not let the reader down. All the old favourites are there along with loads of game reviews.

Being a new magazine the game reviews are of newly released games (over the last 5-10 years at least) although there is a look back older classics.

The styles of the reviews are just like the original in both layout and writing style. Each game has a score along with different opinions from other people.

Games are just a part of this great release though, Lloyd Mangram returns for some letter answering and even the Adventure Trial

get a few pages to review newer romps.

There are also special features covering the Spectrum Next, New hardware, Spectrum art and even a feature on how the original Crash magazine was put together all those years ago.

Crash Smashes make a return, with several games getting this accolade, and to my surprise the strategy section is also present.

The Hall of Slime, playing tips, game maps and developer diaries make up the rest of the content.

As with the original, there are plenty of adverts, and this makes it feel more authentic. Crash was full of adverts for all kinds of Speccy related things, from games, hardware, services and other magazines.

This brought back many happy memories of rushing back from the shop, clutching the new issue, running upstairs to my bedroom and pouring through every page.

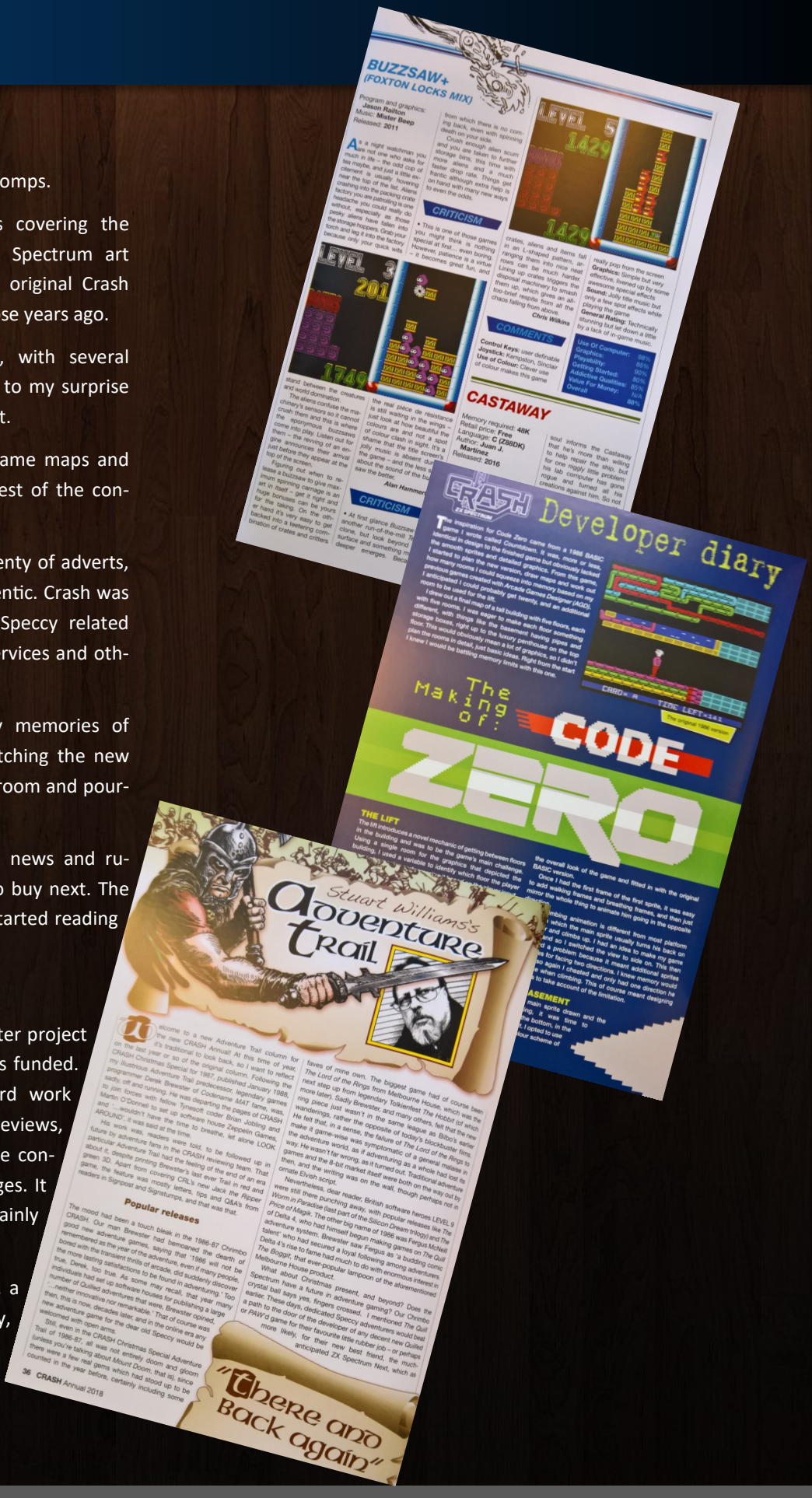
Checking out the game reviews, news and rumours, and making plans what to buy next. The same feeling came back when I started reading this - incredible.

Yes, Loony Jetman is back too !

This book began life as a Kickstarter project and it was no surprise that it was funded. That though was when the hard work began for Chris. Collating all the reviews, producing the layouts, editing the content and wrangling all those images. It was a superb effort that has certainly paid off.

A fantastic piece of history then, a return to the heyday of the Speccy, and one definitely worth buying.

Buy it now.



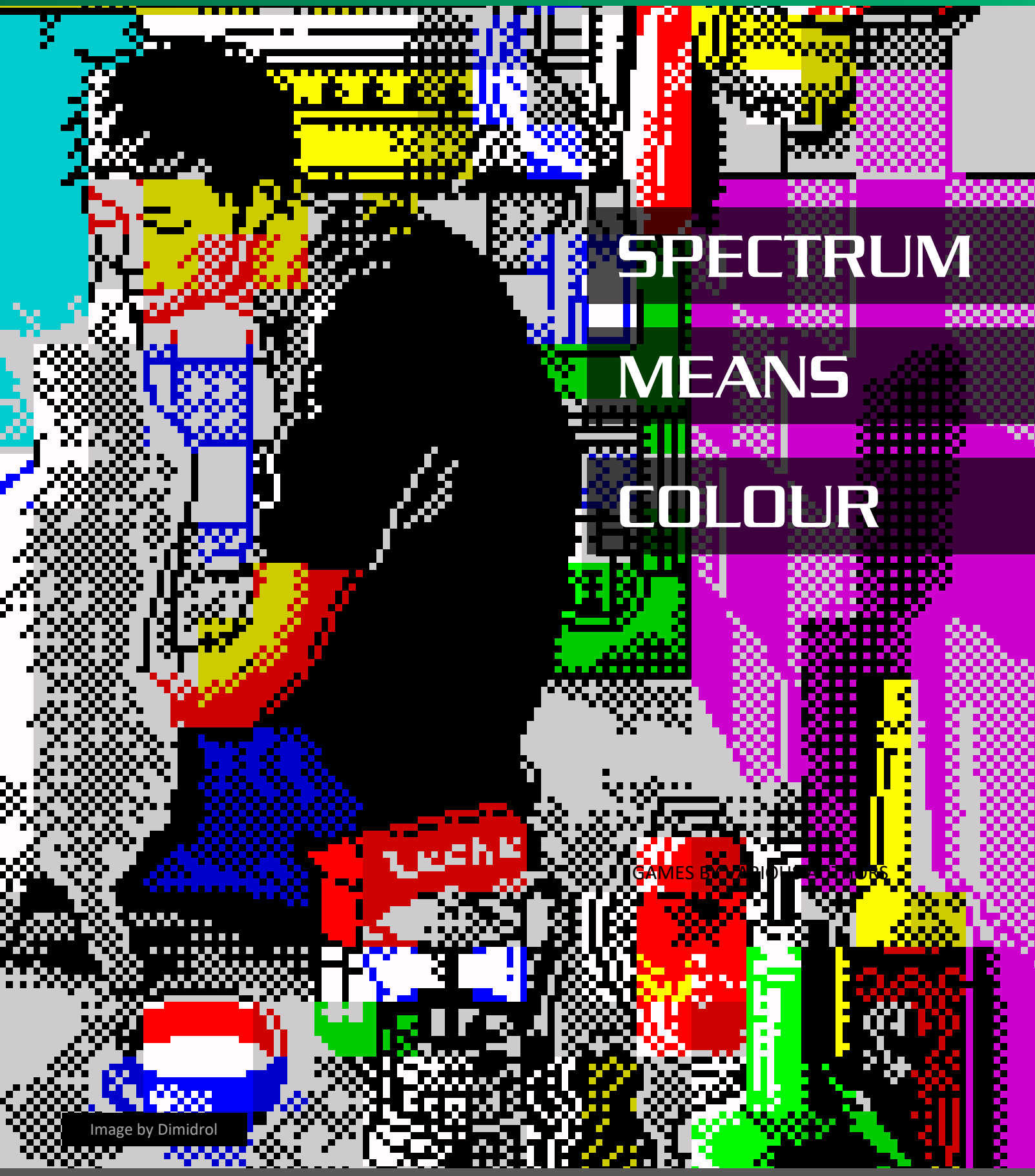
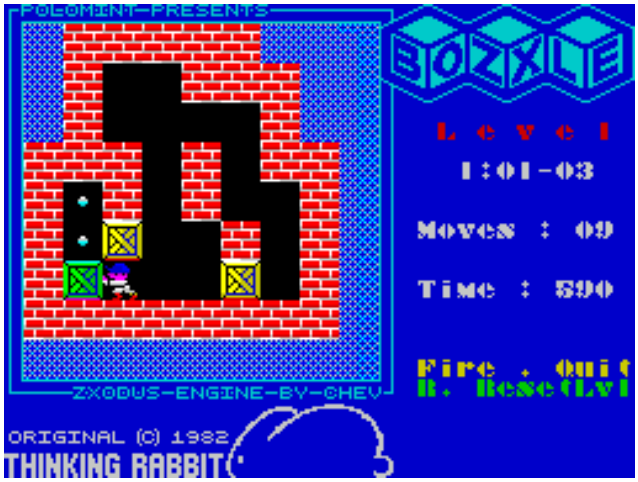


Image by Dimidrol

THE FINAL PART BY PIOTR SZYMANSKI

Welcome to the third and last part of journey through colourful Spectrum games.



BREAKING THE HARDWARE LIMITS

The hardware colour limitations can be overcome by a skilled programmer. There are many Spectrum games where you can see more than 2 colours inside a character square. This effect, called rainbow graphics, was often used for ornamental purposes on the title screens or high-score tables. Things changed in 2011 when Andrew Owen developed the ZXodus engine. It allowed you to display multicolour graphics (an attribute for each pixel line) on an area of 18 rows and 18 columns. This engine was used in a game Bozxle based on Sokoban.

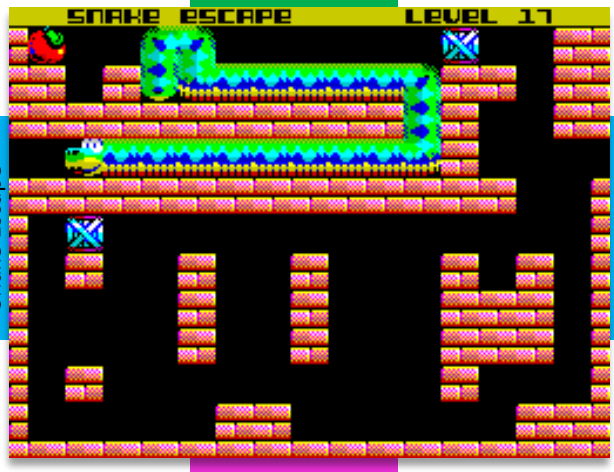
In 2012 Einar Saukas developed the BIFROST* engine. It was compatible with ZXodus but it was also faster and allowed you to display multicolour graphics in 22 rows and 18 columns (BIFROST*2 released in 2016 enhanced this to 20 columns). BIFROST* was used in several games for example Buzzsaw+, Knights & Demons DX and Complica DX.

Dreamwalker

The next step was the Nirvana engine, developed by Einar in 2013. It was bicolour engine (an attribute for every 2 pixel lines) and allowed you to display rainbow graphics almost full screen: 22 rows and 30 columns. Nirvana+ released 2 years later enhanced available screen area to 23 rows and 32 columns.

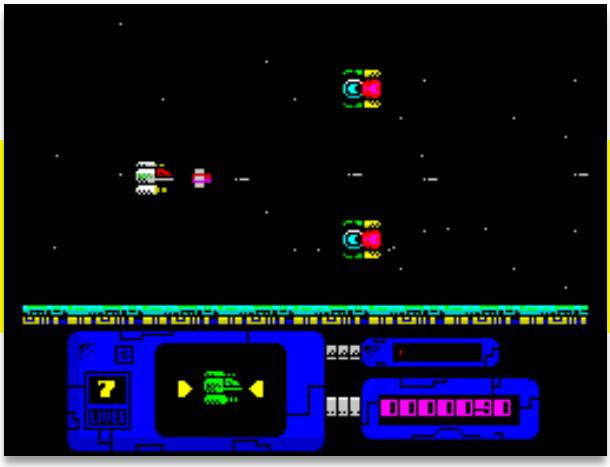
The Nirvana engine was used in Dreamwalker, El Stompo, SunBucket, MultiDude and Stormfinch. There are also 2 games using Nirvana+: Snake Escape and Pietro Bros.

SunBucket

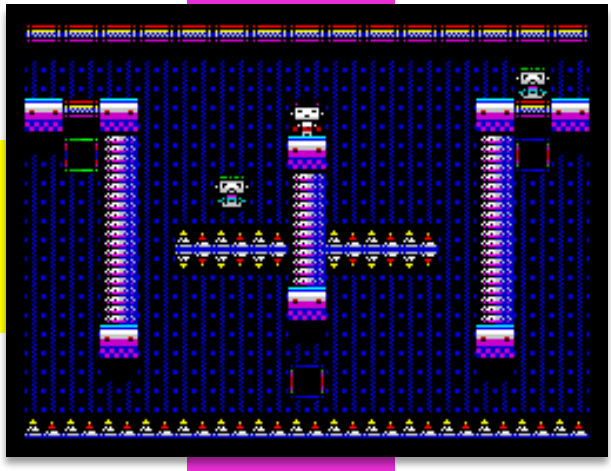


Snake Escape

Stromfinch



MultiDude



WHEN COLOURS DON'T LOOK GOOD

Colourful Spectrum graphics are often pleasant to look at. Often but not always. Sometimes authors try to make their games as colourful as possible but the result is a colourful mess.

Probe Software did an excellent conversion of Turrican but the sequel was converted by an other team, Enigma Variations. It wasn't a good choice. Graphics in Turrican II are colourful but they hurt player's eyes.

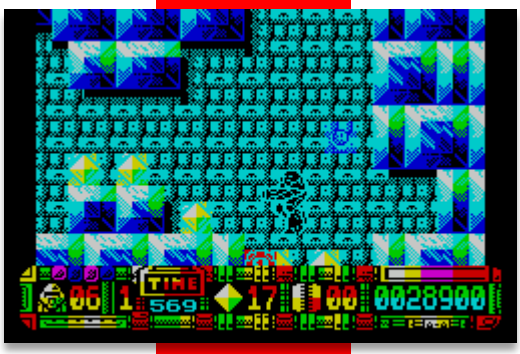
Merlin by Mike Westlake at the beginning looks great. The main character is big and colourful, background graphics are detailed, animation is quite smooth. But after a while you'll notice that Merlin and other sprites are transparent and blend with background objects. This "feature" spoils the fun and it's also present in later Mr. Westlake's games Pieces of Eight and S.A.S. Combat Assault.

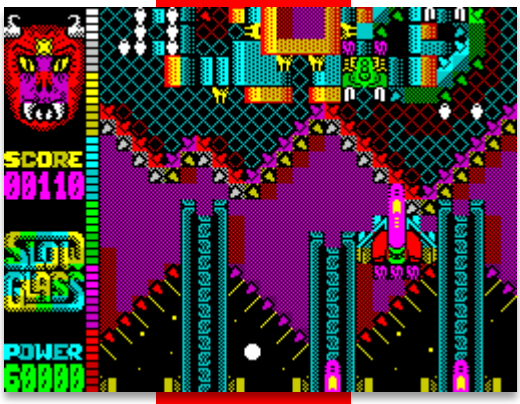
In 1984 Thor released 3 games based on a fable: Jack & The Beanstalk, Giant's Revenge and The House Jack Built. All of them have colourful but also amateurish and ugly graphics. It seems that the authors, Steve & Chris Kerry, were just learning how to write a good game because their later titles are much better.

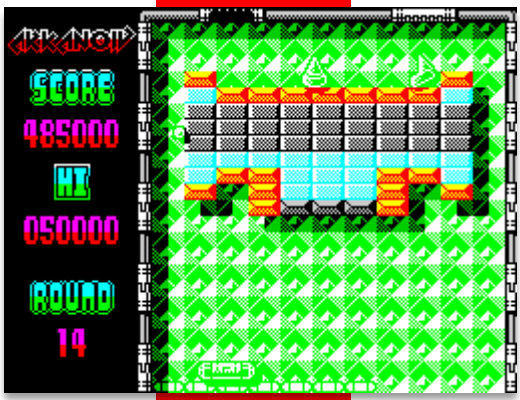
Slowglass is a Spanish shoot'em up by Manuel Dominguez and Alberto Perez. This unreleased (but available to download) title is a perfect example of a game with too colourful graphics overloaded with details. It's really hard to enjoy a game when you can't see what is going on.

In the last few years The Death Squad released many games for Spectrum. They are easily recognizable because of their graphics style which sometimes looks good but sometimes it doesn't. Willy The Wasp 2 belongs to the latter category - the graphics are garish and ugly.

Arkanoid 2 by Imagine is one of the best Breakout clones. The graphics were improved since the first Arkanoid but a few levels are very difficult to play because of the chosen colours. Those levels look better on a black & white TV - maybe the game authors were using this type of TV during development?





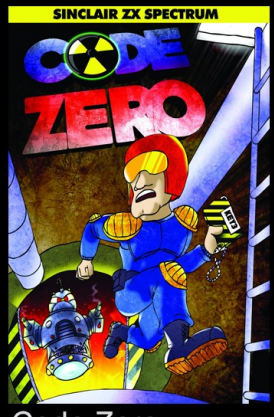


SUMMARY

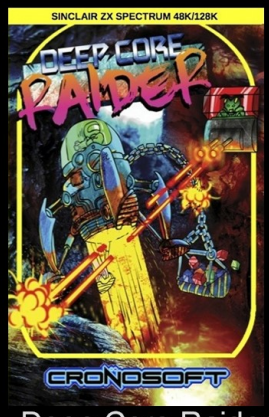
There are, of course, many more colourful Spectrum games but it's almost impossible to list them all (and you probably would be bored with the endless list). I hope that the chosen examples are enough to prove that Spectrum games don't mean monochrome games.

NEW RELEASES ON REAL MEDIA

Toofy In Fanland+



Code Zero



Deep Core Raider+

ALSO AVAILABLE

Games from Paul Jenkinson

Bounty
Test yourself with this sc-fi text adventure.

Baldy ZX
Unique plaform game across 20 levels.