

1X SHED

RETRO SPECTRUM ACTION

A World Of Spectrum Publication
Yeahthatsright! Were Now Officially WOS....ish

ISSUE 3
£Free

Like We'd EVER
STOOP as low
as to BEG for
Donations

THIS ISSUE

GAMEX
IN REVIEW

KEVIN TOMS
INTERVIEW!

AND THE
ODD DALEK
OR TWO



How can 8bit games possibly be unfashionable?



CRONOSOFT

RETROGAMES, Freiston Enterprise Park, Priory Road, Freiston,
Boston, Lincs. PE22 0JZ-UK

WWW.CRONOSOFT.CO.UK ~ ~ chaosmongers@yahoo.com



WELCOME

In true Speccy tradition, Issue 3 is here, and only slightly late! Like about 5 months late...

But what an issue! We have not one, but two tutorials in addition to the excellent code shed articles. Plus all the usual goodies.

And it's been a great year for games too – Jonathan Cauldwell has been his usual rampant self, banging out cracking new titles in his sleep, and of course now we have Bob Smith back in full creative flow. But the big news is that Joffa Smith is back on the scene and announced that he is working on a new title. Keep an eye out for the next issue where we will be reviewing the new game as well as some other goodies from Joffa. In addition, we can now tell you that Firefly II is also planned for next year. The current working title is "Buzz saw". Think large robotic spider/crab - et al.

As for other happenings – there was the retro-fusion event that was attended by my good self. I went along to grab interviews, photos, etc. Unfortunately, things didn't work out that way. Well, you see, there was a bar. Things were going well until a certain member of the Retro Gamer team decided to buy lots of drinks. It's not fair to name him, so I'll refer to him as Martyn X. The reputation of ZX Shed was at stake and large amounts of alcohol were consumed. On the plus side, ZXS definitely won.

Work on the website continues and will be live soon, and the ZXS team are also re-vamping the Cronosoft site! (Are we good or what?)

To top it off, ZXS will be released on a more regular basis! So enjoy, and don't forget to leave your comments on the WoS forum...

Lee Fogarty
Editor

Credits

Editor
Lee Fogarty

Art Editor
Melissa Ward

Additional Design
Sarah Hughes

Main Contributors
Richard Chandler
Skarpo
Ian Hamilton
Steve Parry-Thomas
Ben Rapier
Paul Johns
Andrew Ryals
Arjun Nair
Michael Bruhn

Additional Contributors
Andrew Owen
Daniel Thorlby
Graham Richards
Jonathan Cauldwell
Matt Barber
The Gaffer
Shaun Bebbington
Mr. Anonymous
Matthew Westcott

**Additional Material By Kind
Permission**

www.worldofspectrum.org
www.the-tipshop.co.uk



Martyn X wonders who nicked his pint.

SPECIAL PREVIEW

REGULARS

REVIEWS

54

Another Jonathon Cauldwell game in the making – Izzy Returns!

Bob Smith returns with Stranded 2.5 Can we help Tysch escape his prison?

Joffa Smiff makes a stunning comeback with Saucer

Paul Dunn creates a tribute to Crash reviews

INTERVIEWS

Ben Rapier quizzes Richard Jordon of Retrospec.....44

The Gaffer performs a very hard to transcribe interview with Kevin Toms.....66

SPECIAL FEATURES

Going Commando.....14
Dr. Who28
Vulcan – the Tunisian Campaign.....46
The making of Haiku 4K74
Troll and a half84

News5
What's going on in the Speccy universe?

How To24
Play CD's on your Speccy

How To34
Create a text adventure using PAW

The Code She Guide to:
BASIC – Part II38

The Test Of Time40

Lost Treasures45

Crap Joke Corner52
Does exactly what it says on the tin

Developers Diary.....58
Find out what went on during Loco-Bingo development

Overrated?62
Were games ever given high scores for no reason? Surely not!

Pokemin65
You are cheating bastards!

The Code She Guide to:
Reading the keyboard.....70

HARP-ing On80
Gaming fun with Michael Bruhn.

Jupiter Ace.....84

Page 6

Atic Atac Remake
Another classic Retrospec remake

Egghead IV.....
Cooking fun.

GameX.....
Loads of classics!

Loco Bingo.....
House!

Farmer Jack in Harvest Havoc!.....
Get orrrff moi laaand!

Spectrum Allstars Volume 1.....
Oh dear...

CONTACT US

Got something to say? You can contact any member of the ZX Shed team at zx.shed@gmail.com

If you'd like to make a submission to the magazine, check out the ZX Shed section on <http://www.zxshed.co.uk>

New TZX revision in sight.

The next version of the TZX format is currently in the works and is being handled by the same people who brought you the original TZX – the team at RealSpec. This time the Speccy community has been more actively involved in the development and after much informative and lively discussion, Lucas of the RealSpec team has decided to take up the development initiative. Good as news as that is, one only hopes it doesn't take as long as RealX to come out!

Jonathan Cauldwell to code a game based on WoS forumers.

It's that man again! After having served us such appetizing games as Higgledy Piggeldy and Amusement park, JC is at it again. This time around, however, the premise of the game is an interesting one – it's going to be based on WoS! It all began when somebody on the forums suggested that it would be nice to have a game based on the lively characters that inhabit the Speccy community. As anybody who's been around on the forums for any length of time will tell you, there have been some rather interesting moments in WoS history and all kinds of colourful characters have come and gone. JC is the man who has volunteered to provide code based on inputs from members on the forum in order to make a platform arcade adventure. So, any artists or musicians out there—get creative!

WoS celebrates 11th birthday!

WoS turns 11 years old on 29th November! It's an incredible testimony to the dedication of Martijn and the outstanding commitment of the co-maintainers – Andy Barker, Steve Brown, Tony Barnett, Paul Hurd, Juan Pablo López-Grao, Philip Kendall, Gerard Sweeney and Lee Fogarty – that WoS is a site like no other on the planet. It's not just an archival site for Speccy related material (though it is *the* site for that!), it's also a place where people from different backgrounds, different experiences, different interests, different everything all come together (cue Beatles) to share their point-of-view, opinions, knowledge and a fair bit of completely useless but engaging banter. It's a home away from home for some. Keep an eye out for some extra special treats on the big day...

Secrets of MEMPTR cracked

Vladimir Kladov, author of Emuzwin, announced on the forums that yet another secret had been prised from the depths of the Speccy plumbing.

In Vladimir's own words on WoS (since we don't understand any of it anyway), "As you may be well know there is a mysterious internal register in Z80 CPU (MEMPTR) which value is set or changed by some Z80 instructions and 2 bits of which are

affecting bits 3 and 5 of the Flags register after the instruction BIT n,(HL). But until these days it was not exactly known which Z80 instructions initialize the MEMPTR and which way they do it. Now this MEMPTR register is totally cracked in result of work of an association of Russian/Ukrainian/Belarusian (and other post-USSR) programmers and engineers. The documentation with the results of those hard works is available by the reference: <http://zx.pk.ru/showpost.php?p=43800&postcount=144> (download memptr_eng.txt there)". Stirring stuff I tell you!

New Speccy game draws rave reviews!

Na Than Assh Antti sprung a pleasant surprise on an unsuspecting Speccy community when he announced the completion of Phantomas: Infinity for the Spectrum. The game graphics and coding has been done by Na Than, while the music is by NAMES. The game itself is a colourful platformer that's extremely polished and fun to play. It can be downloaded for free from <http://cezqs.computeremuzone.com/eng/card.php?id=10>. Don't forget to give feed back to the author to let him know how much you liked it! Hope we see more games of such quality in the near future!

Total Eclipse remake is hot!

Smila & Ovine Boyo (names changed to protect the guilty one presumes) has completed their remake of this classic exploration and puzzle solving game and boy, have they done a good job of it! The game features some great looking graphics and sound and is a must play for anyone who liked the Freescape games that were groundbreaking acts in their own time. You can download the game for free from www.retroremakes.com. The author is planning to remake the other Freescape classic that started it all – Driller and we can't wait!

The ZX Spectrum coffee table book is ready!

That's right! Andrew "Don't call me Andy!" Rollings has announced the completion of his epic chronicle of Spectrum classics otherwise known as The ZX Spectrum Book – 1982 – 199x. Those of you who pre-ordered it can congratulate themselves on a good investment since not only is the book choc-a-block with great content but also because Andrew has managed to get the great man Sir Clive Sinclair himself to pen a foreword to the book. Is that great or what? Yes, we thought so too, which is why we plan to be nice to the postman for the next couple of weeks (months?). It would be interesting to see how Andrew has managed to do the balancing act of reviewing the hundreds of great games that came out for the Speccy, considering he'll have to leave out some of the favourites due to constraints of space. With any luck, we'll have something on the released book in our next issue (yes, there will be one).

Atic Atac

Publisher: Retrospec
Release date: 18th May
Price: Free



How it was... the game we all know & love.

BEN: I have a confession to make, so promise you won't laugh? Now, I know I can't be the only one and we're all friends here so that's why I don't feel silly for saying this. When I first played Atic Atac back in 1983... it scared me. Now stop laughing, I must have only been about six years old. But it did scare me quite a lot actually. I remember looking at the cassette inlay and not liking the look of that ghost and as for the thing in the trapdoor, well, I'd rather not think about that.

Loading the game was also a bit of a scary experience too. It was the loading screen you see. The cassette inlay was a photocopied affair (ahem) and I just wasn't prepared for the image in colour (those eyes!). To take my mind off the horrors of the cassette inlay and the loading screen (they're looking right at me!), I read the instructions and the story. I don't care what anyone else thinks, some of the story still gives me the chills today. Especially this bit...

Footsteps... Something's coming down the corridor... Something large... Something cold... I must run quickly... QUICKLY.

And then there was the game itself. Dashing through the rooms, corridors and cellar of the haunted castle as the wizard, the knight or the serf. Avoiding or dispatching the spirits, collecting keys, food and other vital objects. Locating the three pieces of the ACG key to unlock the main doors in the start room and get the hell out of there. Back then I regarded Atic Atac as something of a rarity.

Not every game released managed to have a real atmosphere about them.

So, I loved it (despite those eyes!) and played it again and again, mainly trying to get further in the game, but sometimes just exploring. So I couldn't help but wonder whether those people at Retrospec could actually make any improvements to the game. Obviously they could add pretty colours and all the bells and whistles that you care to mention but would it improve the game? I downloaded it to find out and I'm glad I did.



Look familiar?

Once the game had been installed, I loaded it up by clicking the familiar looking key icon that had been placed on my desktop. A brief, but attractive Retrospec logo popped up and then I was treated to what has to be one of the best introductions to a remake I have ever seen. As the atmospheric melody played and the cartoon style graphics illustrated the back story that unfolded, I found myself asking the question, "Did I just download this... for free?" In my opinion, the back story is what the original game lacked and it's good to see that the team took their time to come up with something that looks and sounds so good. It's funny too and says to me that they wanted this game to be something special from the outset. Was I disappointed that the loading screen with its spectres was gone? Not in the slightest, the intro is great and they've even managed to keep me happy by including... those eyes!

The introduction gives way to the title screen. From here you can start the game, redefine the keys, and mess with some of the options. You can select to play the game in full screen or windowed modes, as well as changing the volume levels and choosing whether you want the text in English or Spanish.

Once you choose to start the game, a final menu gives you the chance to choose your hero. Will you be the knight, the wizard or the serf? Just like the

original, your choice will affect the game as the castle layout will change; each character has a different weapon and can only use the secret passages that they know about. Once you've decided, the game starts.

The objective of this arcade adventure game remains the same. You have managed to get yourself locked in a haunted castle and need to explore its many rooms to find the three pieces of the ACG key. In order to do this, you need to pick up other items and carry them with you to help you on your way. For example, you can't get through the locked blue doors without first opening them with the blue key. One room has a pretty persistent vampire that follows you around... but didn't you see a crucifix in some other room? Once you have located each of the pieces of the key, you need to get back to the start room and open the door and run for your life.



Uh-ho – energy is running a bit low there!

The first thing that hit me about the game itself is that although everything has been updated the game manages to retain all of the charm of the original. The best way to describe it is like this, it's as if the team got the balance of familiar and new ingredients just right. They knew what we loved and kept all that stuff in but also saw exactly what could be changed without marring the game. Play the game, listen to the supernatural movie style soundtrack, then lose a life and you'll see what I'm talking about. The 'get ready' sound is still there and that is something that has always stuck in my mind from the original. Once you start playing you'll hear all of the original sound effects plus many more. The graphics have been given a wonderful make over too.

The rooms and sprites are all presented in full colour but every single one is recognisable as it's

counterpart in the original game.

I don't want to go into too much detail as to what has been updated and improved as it would spoil the surprise. And there are lots of surprises in this game. I only discovered one feature while I was writing this review and I suspect there are others that I haven't encountered yet or didn't notice as I dashed around the castle looking for keys, spanners and crucifixes (oh yes, the vampire, and Frankenstein's monster are present and even scarier than before). At first I found myself running from one room to another and wanting to go back to the last room to see whether I really did see what I thought I saw.



Woo – scary pumpkin

It's hard to fault this game. It really is. It's obvious that the team, led by coder Richard Jordan, with John Blythe giving us the pretty pictures and Matt Simmonds providing the music, thought enough of the original to produce a remake of this calibre. Perhaps they could have added a few features that weren't in the original. A difficulty setting perhaps, and a save game feature would have been nice. But then, would it have been the same game? What I didn't like in the remake was what I was never that fussed about in the original.

All in all, this is a beautifully presented game that has retained all of the best bits, all of the replay value and all of the (ahem) spirit of the original.

Marks

Playability - 9

Graphics - 9

Sound -9

Instant Appeal - 10

Addictiveness - 9

Overall – 92%

REVIEW

Egghead IV

Written by Jonathan Cauldwell

Release Year: 2006

Publisher: Jonathon Cauldwell

Price: Free

LEE: We all know that sequels are generally bad news. Police academy anyone? One sequel after another trying to cash in on a name. Every now and then, something breaks the mould. Take Lethal Weapon as an example.

So, where does Egghead IV fit into this? It's a Lethal Weapon. Jonathon has taken the character further than before whilst still maintaining the fun elements that made the previous Egghead games so playable. In his new venture, our oval friend can pick up and drop objects as well as climbing ladders. This offers a whole new dimension to the character.

So, what's it all about? Following his adventure in space, Egghead is today entertaining a guest. Yes, Shelley is coming to tea and it's up to Egghead to get the food on the table by 3 o'clock. His sentient cooker suggests a recipe for the occasion but it's up to the oval fellow to locate the ingredients dotted around the town where he lives - and that's where you come in.

Sounds simple enough? The game has a very easy learning curve; the ingredients initially seem to be relatively easy to collect. But wait! There are hidden sections to the game that can only be accessed with certain items.

best in-game tunes you will hear. There is a sacrifice in that there are no spot-effects, but it's hardly noticeable. The game itself is extremely well thought out with more examples of Jon's wit shining through. From 'burger Kong' to the method of filling up the jug (eewww). The ingredients are random, potentially making a different game each time.

Anyone living in the Nottingham area will recognise many of the screen locations. A large number of which seem to be drinking establishments! Another nice touch is the pathetic attempt to bribe the editor of this very magazine by featuring a caricature of him in his very own screen. Hmmm - maybe a spin-off game featuring yours truly?



Who's that good looking chap?

Any gripes? Well, once you get into the game, it seems to be quite small. That's more of a testament to the quality and playability of the game though. Maybe an option on the menu to set the number of ingredients required? By having to collect them all you have a potentially difficult game. Also, the timescale seems to be over-generous. Other than that, we have an excellent 2D platformer that easily compares to the speccy's heyday.

Quite simply Jonathon Cauldwell at his very best. A very well deserved 90%.

Marks

Playability - 9

Graphics - 9

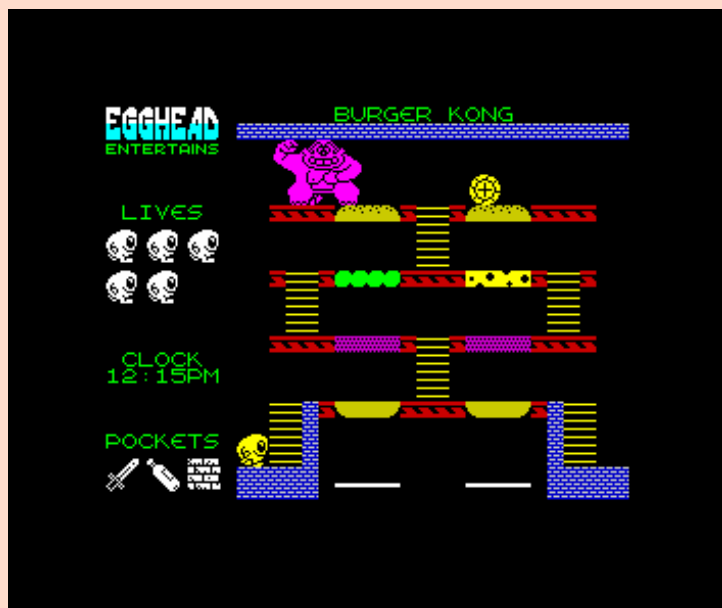
Sound - 10

Instant Appeal - 8

Addictiveness - 9

Overall - 90%

You can download the tap image for free from <http://members.fortunecity.com/jonathan6/egghead/id7.html>



Ha-ha Burger Kong. How clever is that?

From the outset you know you're in for a treat. The sound track by Yerzmyey is quite simply one of the

Gamex

Written by Jonathan Cauldwell

Release Year: 2006

Publisher: Cronosoft

Price: £2.99 (Cassette)

Richard - The one man Spectrum games industry has been at it again. But time he's really gone too far. He's released 16 games all in one go and to make it even more impressive all in one game!! The latest creation from Jonathan Cauldwell's magic keyboard is called Gamex and is Jon's take on the stock market. The premise is very simple, you play a simple Pac-man clone to build up points and avoid losing lives to avoid being taxed. Once you have cleared the screen you can use the points to buy shares in any of the 16 games listed on the right hand side of the screen. Once you've allocated your points to games you then play the games to get as high as possible on the scoreboard.

Admittedly they're not the most complex games you'll ever see but Jon's only got 48k to play with after all so don't expect to see an Elite or Starion clone amongst them. On the other hand the majority of them are surprisingly addictive with a few well known favourites lurking in their midst. Missile Command, Moon Lander and Pac-man are just three you'll find but I won't reveal anymore as part of the fun is discovering the new games.

Once you master the games it becomes a strategy game where you have to maximise your use of points to ensure you get as many games as possible in your portfolio which will allow you in turn to build up more points for the high score table. Personally I never worried about that and just enjoyed discovering all the games and pretty much marvelled at the fact that Jon managed to squeeze 16 games into 48k.



This is a game (games?) which has all the best attributes of a piece of Jonathon Cauldwell software. Clear, colourful, neatly laid out screens, smooth sprites and pleasant unobtrusive sound effects. Jon's games are never technologically groundbreaking (though 16 games in one is pretty impressive) but they do have that addictive factor that is so much more important than flashy graphics and thumping soundtracks. Given that I'm surprised no-one else is really producing anything of similar quality or does Jon just have the magic touch? In light of recent events it appears there could be some money in new Spectrum games so if you've even felt you could write a game of a similar standard then I would suggest that now is the time to do it. Retro is not dead and anyone who says so is plainly talking soft.

Marks

Playability - 9

Graphics - 9

Sound - 9

Instant Appeal - 8

Addictiveness - 8

Overall - 83%

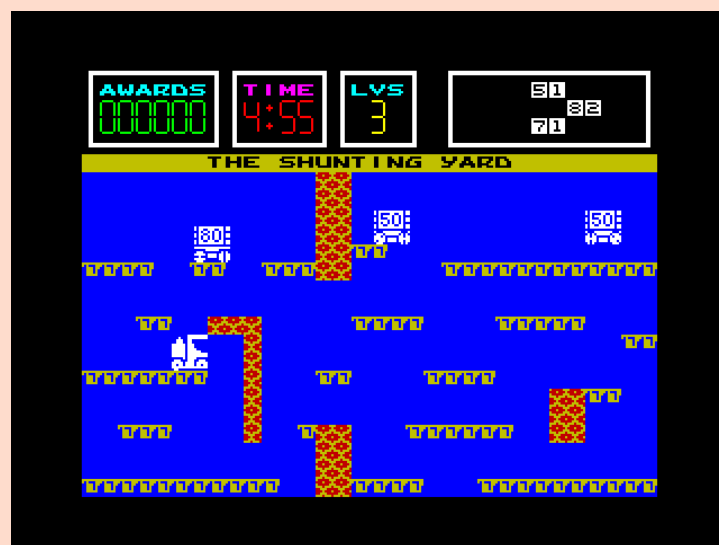
Loco Bingo

Written by Jonathan Cauldwell

Release Year: 2006

Publisher: Cronosoft

Price: £2.99 (Cassette)



In the future, a brand new (and highly unstable) chemical element is discovered and is quickly found to be useless as it is prone to exploding. The company that produces it has to get rid of it quickly so what do they do? They sell it to a company that runs TV Bingo games. Chug Horncastle, a sentient train (?) helps has to move numbered crates containing the new element around a series of

REVIEW

platforms and combine them to form the numbers on his own bingo card. Lucky him! In the real world, people (and trains, I assume) that work on those sort of shows are automatically exempt from entering.

A game with a plot as strange as this could only be successfully pulled off by one of two people - Matthew Smith is one and the other is Jonathan Cauldwell. It's an old cliché but I'm wondering if a few joints or some LSD played a part in the creation of this game, it's that bizarre.

Graphics, as always with Jonathan's games, are excellent. There is no colour or attribute clash and movement of the sprites is extremely smooth. Sound is absent, from the 48K version anyway (I've been unable to test it on a 128K machine as mine is currently acting up).

Now we come to game play. This game is so addictive it should carry a government warning. When I first got it, I played it for two hours straight and that was a 3 O'clock in the morning. Basically, you have to push the crates around the platforms and combine them so the number matches what's displayed in the status panel. It's not too difficult at first but it does get much harder on later screens.

Jonathan Pryce once said (in Tomorrow Never Dies) "The distance between insanity and genius is measured only by success." Let's hope this game is a success.

Marks

Playability - 9

Graphics - 8

Sound - n/a

Instant Appeal - 9

Addictiveness - 10

Overall - 90%

Farmer Jack in Harvest Havoc!

Written by Bob Smith

Music and Sound Effects Lee Du-Caine

Published by Cronosoft

£2.99 (Plus 99p P&P)

Available Now

Coming soon on Microdrive



Now Loading

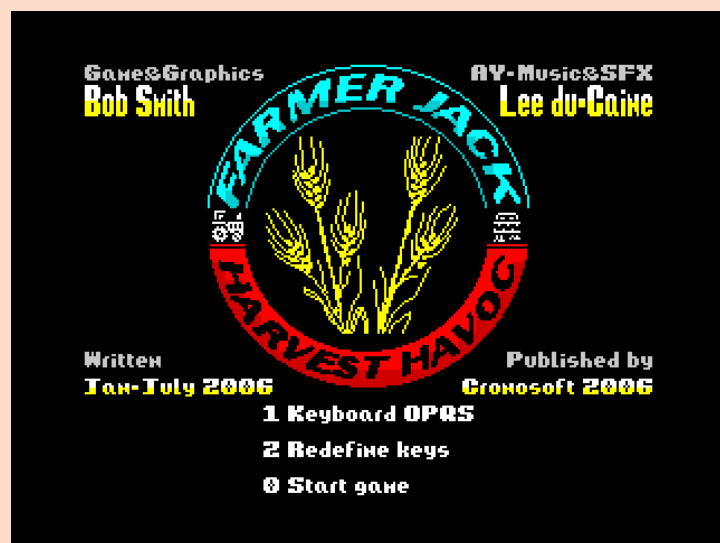
Melissa: You are Farmer Jack. A man who does love his home brewed wine! From a small hobby Jack has turned his passion for home brew into a fully fledged business. A business that just gets bigger and bigger every year until now, finally, his quality organic produce is number one throughout the land and enjoyed by millions.

Naturally an operation like this will draw a great deal of attention. And not all of it good!

Late one night, just as his crops are bursting and thoughts of the harvest firmly in mind, Jack hears the strange sound of clanking and grinding metal coming from his prize winning fields of produce. Nervously he draws back his curtains to see a hoard of giant mechanical menaces, sent by his evil competitors, trampling over his livelihood!

"By eck! Me 'arvest! It's a rum-do this!" Jack exclaimed and bounds to his faithful tractor and races off to collect his harvest before it trampled into oblivion!

And so begins Farmer Jack in Harvest Havoc!



Let The Games Begin!

Every now and then a program comes along that reminds you just what it was that was so god dam amazing about the ZX Spectrum and it's games. A game that grabs you by the lapels and screams in you face reminding you how utterly awesome the Spectrum was and still can be. A game that has that one, key, critical ingredient that keeps you going back and back and back for more. **Farmer Jack in Harvest Havoc!** is such a game.

First of all we need to point out the obvious just to put it out of the way. Farmer Jack is a 'Dig Dug' clone, a Dig Dug clone with a dash of 'Bomberman' thrown in for good measure.

So where does that leave us? Well, for starters, one

thing the Spectrum really needed back in the day was a REALLY good 'Did Dug' clone. There wasn't one, well; there wasn't a quality one... until now. So, now we've put that to bed, what about the Game?



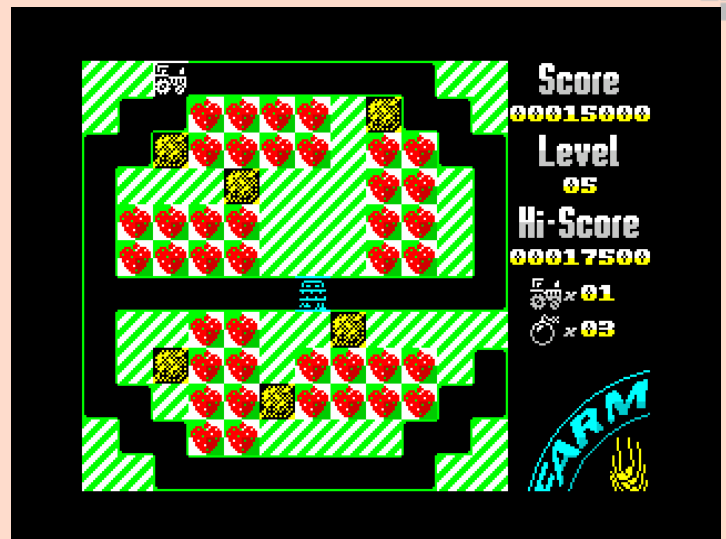
Level 1



Level 4

Each level represents one of Jack's allotments, full of apples, strawberries, bananas, even the odd cherry. In the centre of the screen you see one of the maniacal munching monster machines that gradually multiply and start prowling round the screen looking for you, chomping anything that gets in their path.

However, you're not totally defenceless. As you plough your way through the field collecting your precious harvest, you can dislodge giant boulders (or are they hay bails?) that will come crashing down, crushing anything nasty that is closing in behind you. But that's not all, you are also blessed with three timed bombs that when released will, after a few seconds blast anything within reach and in all four directions! (Including you if you're not quick enough, so be careful with them!)



Level 5

The level of difficulty is very well balanced too, the first few levels are pretty easy, well, easy enough for you to get a feel for the game without losing too many lives, when you get to level ten things start to get noticeably faster and those little blue buggers start getting a lot smarter too.

As for the controls, I'm not a joystick user, so I was playing with redefined key set. The controls can only be described as slick, smooth and completely responsive. No annoying jerks, just solid playability that makes the experience a lot more enjoyable.

Then we come to Lee Du-Caine sound effects and score. Again, flawless. We have a wonderful array of sound and spot effects that just bring the game to life, along with a cheerful and extremely well composed theme that plays constantly in the background, without becoming annoying or grating like so many an in game tunes can.



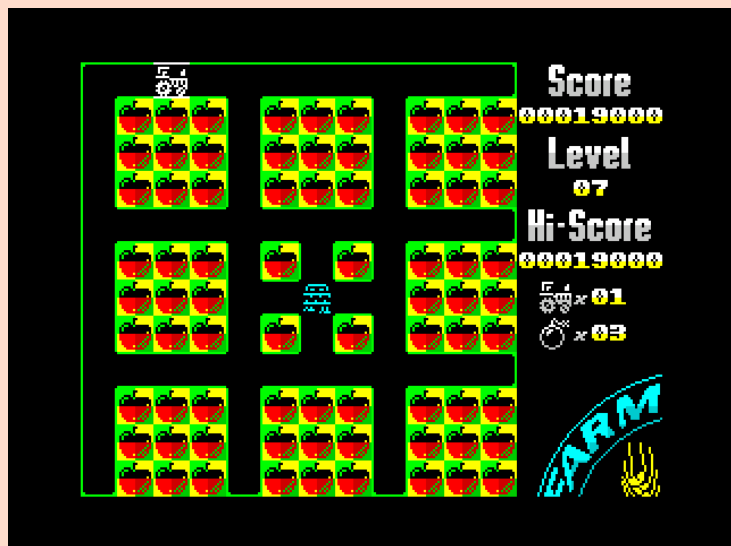
Level 3

Finally, the program is polished off with a fun cartoon loading screen, and a smart, crisp options

page. Perfik!

If I had to make a criticism it would be the player character, Farmer Jack's tractor. It's a basic 2 frame bob animation that works reasonably well, but I would have liked to have seen a front and back view of it for the up and down motion, not just the same graphic rotated 90 degrees. But hey, I'm nitpicking here! Yes, I have found it that difficult to find anything here to criticise this game.

Whilst playing it I found myself thinking about the time when I played 'Bomb Jack' on the Spectrum for the first time, this games playability grabbed me in the same way Bomb Jack did all those years ago, in smooth control, appearance and addictiveness these two games are very much alike.



Level 7

In conclusion, Farmer Jack is a modern day ZX Spectrum classic. I kid you not. It is extremely playable, fun and addictive. Everything a good game should be. I have no doubt that had this game been released in 1984 at £2.99 you would probably be looking at a Crash Smash. Yes, I do think it's that good. But don't take my word for it, Buy it, play it. You will love it. Especially if you're a fan of good old fashioned, classic style retro arcade action! It's the mutts!

Marks:

Playability - 10

Graphics - 7

Sound - 9

Instant Appeal - 8

Addictiveness - 9

Overall - 93%



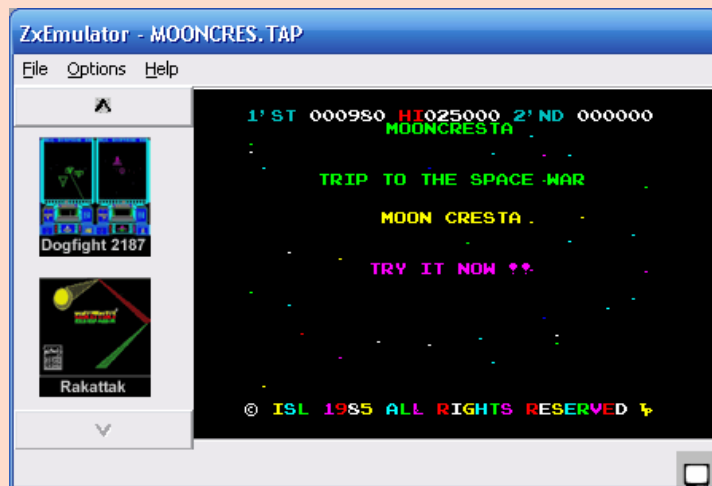
If you've never bought a Cronosoft game before. Buy this one.

Spectrum Allstars Volume 1

Published by Alten8

£4.99 (Plus 99p P&P)

Available Now



Lee: It's a tricky one. Do I review the emulator or the games?

Unfortunately, the emulator lets the whole package down. The fact that it is version 1.0 says it all – was it tested on multiple systems? On installing, it asks you to reboot – for no reason. I chose not to reboot and still got a fully working program! So, I get this box that takes up less than 1/6 of my screen. I am presented with a scrolling box that shows 2 very small screenshots of available games.

Considering there are 50 games, showing 2 at a time seems tedious. Plus, because of the size of the

images, it is sometimes tricky to see what the game is. So, I press the 'full screen' button. This gives the game full screen, but not the menus. It also screws up the display on my 2nd monitor. The games themselves are ok – certainly NOT Allstars. When you think of Speccy Allstars you think of Ultimate, Elite, etc., not a YS type-in! There are some classics there such as Ant Attack, and many of Jonathan Cauldwell's titles, but they are not enough to pull this off. Another gripe is that there are no instructions for the games – surely that wouldn't have been too difficult to implement? Overall, the whole package is shoddy and seems to have been rushed out. We all know that writing a decent emulator takes time, but this one doesn't need all the bells & whistles – it just needs to be able to play the games. In that respect it works – barely.

The games make up the score here – 50%.



Shaun: Alten8's Sinclair ZX Spectrum All Stars Volume 1 may have many short comings. For instance, on the Windows 98 machines I tested it on, I couldn't get the sound to output, not even the beeper, and I don't like the fact that the window that the emulator runs in can not be scaled to any size. So, the higher the resolution you run your Windows PC, the smaller the emulator will be on your desk top. There is a full screen mode, but I found that the border wasn't emulated, and on "Wide Screen" monitor, you will find that there is a huge chunk of black on the right hand side.

I could go on, but, on balance, there are some positives; the first being quite close to my heart and that is the collection contains many home-brew Speccy games. Hopefully, this will mean that the more casual Speccy fan will be drawn into one of Cauldwell's great games. But it's a mixed bag of good and bad – let's hope that they avoid the likes

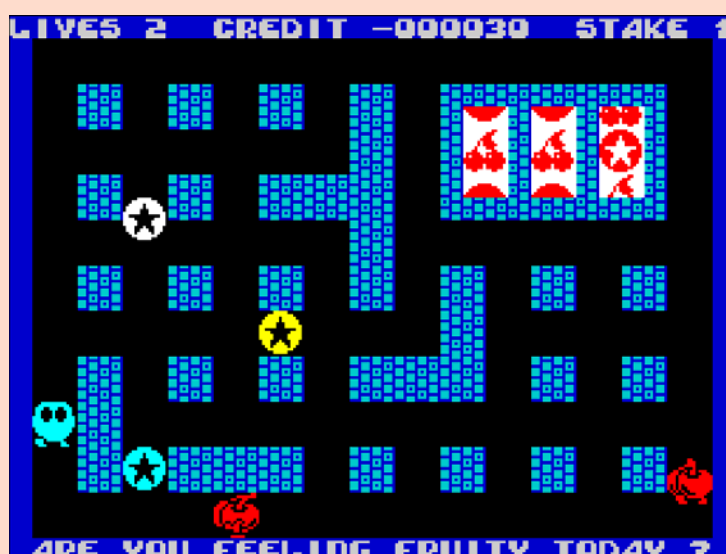
of "Hang the Man".

Knowing that any proceeds made from this collection will be ending up in the pockets of Mr Cauldwell, who creates his wares out of love and not for profit, brings a smile to my face.

There are also some quality older games to play, which should keep the nostalgia hunters happy. 3D Ant Attack, Bloodwych and Shark! spring to mind here. However, there seems not to be any instructions included for the games. WoS is your friend here, but would the casual Speccy fan know this?



Overall, the collection of games in on the whole good, while the emulator seems very lacking, but at least you can load the games into any emulator you have to hand.



And it is the games that mark up my final verdict; 66%

NEW FEATURE

Going



Following on from his excellent Bomb Jack editor, Mr. Anonymous has been at it again! This time, the classic Commando gets his treatment. In the first of two parts we show you how to edit the game.

Making levels in Commando is easy, real easy. Even the most novice of Spectrum users can have a go.

The good news is that we don't have to worry about the enemy soldiers. They will interact with the scenery automatically. For the majority of the time, all that is required is for you to look up the code for the graphic you require (see table below) and decide which column to locate it. There are some small exceptions to the rule however, such as having to slot the different components that make up bridges and rivers together with predetermined amounts of space.

Have a quick browse at the graphical table below and we'll set about writing our first level. Look up the value for a tree and a single grenade. The graphical code for a tree is \$83 and for a single grenade it is \$0F.



Level 1 before we start

We will place the tree in column 10 (ten) and the grenade in column 1 (one). In hexadecimal notation the equivalent of 10 is \$0A and 1 is \$01.

Let's put them into place then:

\$83, \$0A ; Here's the tree in column \$0A
\$0F, \$01 ; Here's the single grenade in column \$01

Putting these items one after another means that they will appear on the same **ROW**.

That's all very well until we want to put objects at different heights on the screen.

In order to achieve this we use Land Space and a quick glance at the table shows the code of Land Space to be \$00. Inserting Land Space tells the program to move up the screen before commencing to draw a new object. How far we move up the screen depends on the value we give to Land Space. The value can be a number from \$00-\$FF (0-255). The smaller the number, the less we move and likewise, the greater the number, the further we move and the more land we have to traverse before reaching the next object.

Examples

\$00, \$42 ; Example 1

\$00, \$CE ; Example 2

\$00, \$6A ; Example 3

Let's update our program then:

\$83, \$0A

\$00, \$33 ; Here's our "space"

\$0F, \$01

This means that there will be a nice amount of walking space between our tree and the next object, a single grenade. In this case it works out to be approximately half a screen worth of space. Now that our program is starting to grow we better start to keep an eye on it so load up your copy of Commando and open your assembler.

I will be using Spin's assembler. EmuZWin has a built in assembler and works in a similar fashion. For other users such as those using Spectaculator I recommend that you first type out the program using notepad and then use PasmO. (Full instructions at end of this tutorial)

<http://www.arrakis.es/~ninsesabe/pasmo/>

It is also worth pointing out at this stage that during testing "graphical corruption" will occur but do not worry. At this stage we are testing the design only.

Now have a quick browse at the 'Levels Memory Layout' table below and you will see that the code for level 1 starts at \$9236.

With Spin, go to Tools>z80 assembler and the assembler will open showing a new page. Now type the following but press the TAB key before entering each line.

Load up your original game of Commando then go back to the assembler window and press ALT+A to assemble the data. If all has gone well, the assembler will report NO errors and you can now start the game.



How Level 1 should look now..

Hey, look at that! There's our tree and grenade but what's this?!? When we get killed the game resets. This brings us to an important lesson about spawn points:

\$22 Spawn Point/Level End

This (\$22) will be followed by either \$00 or \$01. \$22, \$00 signifies a spawn point i.e. after death you will regenerate at this point. \$22, \$01 signifies that you would like the level to end and fight the last remaining barrage of soldiers.

An important point to remember here is that ALL new games must start with \$22, \$00. Upon death the program backtracks to the last instance of \$22 and then adds two so it knows where to restart from after you lose a life. At the end of a level, the last piece of code will always be \$22, \$01 and thus adding two here will always point to the start of the next level.

However, at the beginning of level one we have no previous 'Level End' code and thus must enter a spawn point first. Failure to do so will result in a crash.

So we can see that we need to start our level with \$22, \$00 and our program now reads thus:

```
org      $9236          ;start of level one
defb     $22, $00       ;this will ALWAYS
                        ;be your first line

defb     $83, $0A
defb     $00, $33
defb     $0F, $01
defb     $00, $FF
```

Another point to remember with spawn points is that if we have an instruction such as the alarm sounding or NME with rocket launchers in action (explained later) then we must initiate these again after a spawn point has been reached.

Next, let's add a sandbag position.

\$8A, \$8B, \$8C Sandbags

A simple case of deciding which column you want the sandbags to appear in. Just take into consideration that the end sections take up two columns and the middle one three columns.

Example 1

```
$8A, $00      ;
$8B, $02      ;a full width sandbag
               ;defensive position starting
               ;at column $00
$8C, $05      ;
```

Example 2

```
$8B, $1D      ;draws a semi width (left and
               ;middle sections only) sandbag
$8A, $1B      ;a full width sandbag
               ;defensive position starting
               ;at column $00
$8C, $05      ;defensive position at the
               ;RHS of the screen
```

Example 3

```
$8A, $00      ;draws a very wide sandbag
               ;defensive
$8B, $02      ;position starting at column
               ;$00
$8B, $05      ;The second $8B won't
               ;contain an NME soldier
               ;though as they
$8B, $08      ;appear only every other
               ;instance of $8B. This
               ;example contains
$8C, $0B      ;two enemy soldiers behind
               ;the sandbags
```

That seems simple enough so we will draw the full width sandbag position to the right of the screen starting in column \$14. Our program now becomes:

```
org      $9236
defb     $22, $00
defb     $83, $0A
defb     $00, $33
defb     $0F, $01
defb     $00, $26    ;I've added a bit more
                     ;space thus
                     ;moving up the screen
                     ;again
defb     $8A,$14     ;draw sandbags LHS in
                     ;column $14
defb     $8B,$16     ;draw sandbags middle
                     ;with NME starting at
                     ;column $16
defb     $8C,$19     ;draw sandbags RHS in
                     ;column $19
                     ;i.e. 3 columns after
                     ;the middle section
defb     $00, $FF
```

Now you've got the hang of sandbags let's add a trench.

\$23, \$24, \$25 Trenches

Exactly the same principle as for sandbags except the middle section takes up four columns.

Example 1

```
$23, $05      ;
$24, $07      ;draws a full width trench
               ;starting at column $05
$25, $0B      ;
```

Example 2

```
$23, $09      ;draws a very wide trench
               ;starting at column $09
$24, $0B      ;this example will have the
               ;full quota of
$24, $0F      ;four NME soldiers as for
               ;every middle section
$24, $13      ;drawn ($24) there will be an
               ;NME positioned
$24, $17      ;there unlike the sandbags
$25, $1B
```

As the sandbags are to the right we will place the trench to the left starting in column three.

```
org      $9236
defb     $22, $00
defb     $83, $0A
defb     $00, $33
defb     $0F, $01
defb     $00, $26
defb     $8A, $14
defb     $8B, $16
defb     $8C, $19
defb     $00, $40    ;Adding a bit more
                     ;space to move up the
                     ;screen again
defb     $23, $03    ;draw a trench LHS in
                     ;column $03
defb     $24, $05    ;draw a trench middle
                     ;with NME in column
                     ;$05
defb     $25, $09    ;draw a trench RHS in
                     ;column $09
defb     $00, $FF
```

Another grenade bonus is required now I think. We will we also add a boulder with NME, a mortar and another tree too. These all follow the same principle whereby you just need to decide which column to place the object in.

```
org      $9236
defb     $22, $00
defb     $83, $0A
defb     $00, $33
defb     $0F, $01
defb     $00, $26
defb     $8A,$14
defb     $8B,$16
defb     $8C,$19
defb     $00, $40
defb     $23, $03
defb     $24, $05
defb     $25, $09
defb     $00, $37    ;Added more space af-
                     ;ter the trench
defb     $10, $0A    ;draws a triple gre-
                     ;nade in column $0A
```

When you test out the above you will immediately


```
defb    $83, $12    ;draws a tree in
                    column $12

defb    $87, $05    ;draws a boulder with
                    NME in column $05

defb    $12, $00    ;draws a right mortar
                    in column $00

defb    $00, $FF
```

notice that the new items are all on the same row. This is because we didn't add any space in between drawing the objects. We only added it after drawing the trench and the next four items appear one after another with no space between. When you want objects on the same row just don't add any space.

Next, we will add a cliff.

\$82 \$26 Cliff and Cliff Edges

In normal circumstances the screen comprises of columns numbered from \$00 (LHS) to \$1F (RHS). When dealing with Cliffs we can also add \$80 onto the column to "mirror" the image.

Note that the last column we can start drawing a cliff (due to the size of the graphic) so that it fits on the screen without overlap is \$1A. This also applies to other larger objects.

Example

```
$82, $10    ;draw a cliff in column $10
$82, $8A    ;draw a cliff in column $0A
            and mirror it by adding $80
            ;Result is a nice big cliff.
```

To complete the cliff we add cliff edges. With cliff edges we use the same system as for cliffs.

Example

```
$82, $10    ;draw a cliff in column $10
$82, $8A    ;draw a cliff in column $0A
            and mirror it by adding $80

$26, $09    ;draw a cliff edge in column
            $09

$26, $96    ;draw a cliff edge in column
            $16 and
            ;add $80 to mirror the image.
            ;Result is a nice big cliff
            with cliff edges.
```

However, when we are drawing cliffs at the edge of the screen it is normal practice that we wish for soldiers to jump off them and attack. To allow this we must make a slight change to the numbers we add onto the column value for the cliff edges. For single cliffs you add \$40 onto the column and to mirror the image add \$C0.

Example

```
$82, $9A    ;draw a mirrored cliff in
            column $1A (adding $80 to
            mirror)

$26, $59    ;still drawing a cliff edge in
            column $19 but now adding $40
```

Or

```
$82, $00    ;draw a cliff in column $00
            i.e. LHS of screen

$26, $C6    ;draw the cliff edge in column
            $06 but mirror it by adding
            $C0
```

Now watch as the soldiers jump off the cliffs to attack.

To allow jumping from double cliffs, a normal and a mirrored cliff side by side, you add \$60 onto the column and when the image is mirrored you add \$20.

Example

```
$82, $80    ;draw a mirrored cliff in
            column $00 (adding $80 to
            mirror)

$82, $06    ;draw a cliff in column $06
$26, $EC    ;draw a mirrored
            cliff edge in column $0C
            ;adding $C0 to mirror then
            $20 to allow cliff jumping
```

Or

```
$82, $1A    ;draw a cliff in column $1A
$82, $94    ;draw a mirrored cliff in
            column $14 (adding $80 to
            mirror)

$26, $73    ;draw a cliff edge in column
            $13 (add $60 to allow cliff
            jumping)
```

I've chosen to include the single cliff that soldiers can jump off and place it at the RHS of the screen. At this stage we will also include the POW bonus.



AREA 01 PART 1

\$20 POW

Every time you want a POW bonus to appear just add \$20, \$00 into your code.

They will always start running from the same position in the centre of the screen.

Placing them in our program:

```
org      $9236
defb     $22, $00
defb     $83, $0A
defb     $00, $33
defb     $0F, $01
defb     $00, $26
defb     $8A,$14
defb     $8B,$16
defb     $8C,$19
defb     $00, $40
defb     $23, $03
defb     $24, $05
defb     $25, $09
defb     $00, $37
defb     $10, $0A
defb     $83, $12
defb     $87, $05
defb     $12, $00
defb     $00, $20      ;add more space
defb     $82, $9A      ;draw a cliff in col-
                      ;umn $1A i.e. RHS of
                      ;screen
defb     $26, $59      ;drawing a cliff edge
                      ;in column $19 but now
                      ;adding $40
                      ;onto the column to
                      ;allow soldiers to jump
                      ;off
defb     $00, $31      ;add more space and
                      ;then
defb     $20, $00      ;include the POW bonus
defb     $00, $FF
```

With the program starting to get rather large now only new code that we will be adding to our program will be shown from this point. All newly added code will be added between the last but one and the last line each time. This is shown by above.

Now we'll add some transport.

\$1A, \$1B, \$1C Truck, Motorbike and Jeep

When you want one of these vehicles to appear, use the following code

```
$1A, $00 ;Truck (The second value is
          ALWAYS $00)
$1C, $FF ;Jeep (The second value is
          ALWAYS $FF)
$1B, $FF ;Motorbike (The second value is
          USUALLY $FF)
          ;The Motorbike second value
          ($FF) can be
          ;substituted for a value $01-
          $71)
```

With motorbikes, a value of \$FF signifies that the motorbike will travel left right across the screen before disappearing from view. You can change the \$FF for a value \$01-\$71 so that it still travels across the screen but stops and remains in view. \$01 is the LHS of the screen and \$71 is the rightmost. \$1B, \$3C will stop the motorbike approximately half way across the screen.

Let's add those into our code:

Next we'll add a barracks and a cave at opposite ends of the screen.

\$95 Barracks

```
$00, $72 ;a nice amount of space to
          allow room for the POW
          bonus
$1B, $FF ;initiate the motorbike
$00, $45 ;some more space
$1C, $FF ;initiate the jeep
$00, $32 ;space
$1A, $00 ;initiate the truck
```

These MUST appear in column \$00 on the left hand side of the screen so the code for the barracks will



always be \$95, \$00.

\$1D, \$1E, \$1F Caves

Simple to draw and to use. We just have to remember to insert a set value of space in between drawing the different cave sections.

Single Cave

```
$1F, $1A ;draw a cave bottom section
$00, $03 ;insert some space. This
          value will ALWAYS be the same
$1E, $1A ;draw the main cave section
$00, $13 ;insert more space. This
          value will ALWAYS be the same
$1D, $1A ;finally add the top cave
          section
```

Multiple Caves

For multiple caves you would repeat adding \$1E, \$1A and \$00, \$13 after the \$00, \$13 above before finally adding the \$1D, \$1A to end the caves.

Example

```
$1F, $1A ;draw a cave bottom section
$00, $03 ;insert some space. This
          value will ALWAYS be the same
$1E, $1A ;draw the main cave section
$00, $13 ;insert more space. This
          value will ALWAYS be the same
$1E, $1A ;draw the main cave section
$00, $13 ;insert more space. This
          value will ALWAYS be the same
$1E, $1A ;draw the main cave section
$00, $13 ;insert more space. This
          value will ALWAYS be the same
$1D, $1A ;finally add the top cave
          section
```



I hope there are no bats in there?

Adding the single cave into our program:

```
$00, $32 ;space
$95, $00 ;draw a barracks in column $00
          ;no space between the barracks
          and cave
          ;signifies that they will be
          on the same row
$1F, $1A ;draw a cave bottom section in
          column $1A
$00, $03 ;insert the required space
$1E, $1A ;draw a cave in column $1A
$00, $13 ;insert the required space
$1D, $1A ;draw a cave top section in
          column $1A
```

The player must be pretty tired by this point so we will boost their spirit with the fact that they have reached the halfway bridge and include that next.

Halfway bridge

You can put as many of these in your level as you wish but the tunnel must always be in the middle.

The code will remain the same, the only exception being whether to include a motorbike riding across it or not.

Example (Without Motorbike)

```
$09, $0B ;draw wall from right to left
          starting in column $0B
$09, $1F ;draw wall from right to left
          starting in column $1F
$96, $0C ;draw tunnel
$00, $17 ;space
$17, $1F ;draw white road from right to
          left starting in column $1F
$00, $0C ;space
$08, $1F ;draw final bridge section
```

Example (With Motorbike)

```
$09, $0B ;draw wall from right to left
          starting in column $0B
$09, $1F ;draw wall from right to left
          starting in column $1F
$96, $0C ;draw tunnel
$00, $17 ;space
$17, $1F ;draw white road from right to
          left starting in column $1F
$00, $01 ;space
$1B, $3C ;draw moving motorbike to posi-
          tion $3C
          ;(approximately half way across
          screen)
```

BEFORE AFTER

```
$00, $0B ;space  
$08, $1F ;draw final bridge section
```

We will use the 'with motorbike' option so we'll include everything printed under the 'Example (With Motorbike)' heading above PLUS an initial line of \$00, \$41 for some extra space before commencing to draw the bridge. Another idea will be to include a spawn point at this point so tag the following code onto the end as well.

```
$00, $55 ;space  
$22, $00 ;spawn point  
$00, $54 ;space  
;take particular note that  
;this amount of space will mean  
;that the  
;next object (log crossing) is  
;not drawn on the screen  
;straight  
;away (see below) and thus  
;will appear correctly.
```

We will start the next section of our level with a log crossing.

River Crossings

There are two options available here. Simply copy and paste the code below for your own level. You can have the log crossing to the right or the log crossing to the left.

The first one you encounter on a level must not be on the starting screen otherwise you won't see the full body of water.

Code for LEFT river crossing

```
$98, $08 ;draw the logs in column $08  
$00, $02 ;land space  
$0D, $10 ;draw the water  
$00, $19 ;land space  
$19, $10 ;draw water strip 1  
$00, $05 ;land space  
$0E, $10 ;draw water strip 2
```

Code for RIGHT river crossing

```
$98, $10 ;draw the logs in column $10  
$00, $02 ;land space  
$0D, $08 ;draw the water  
$00, $19 ;land space  
$19, $08 ;draw water strip 1  
$00, $05 ;land space  
$0E, $08 ;draw water strip 2
```



EEK! I haven't got my trunks!

Decide which type you would like then copy the code into your program. After negotiating the log crossing we will up the tempo with an attack by rocket launcher wielding soldiers. Whilst the attack is taking place we can still add other items onto the map as desired. In this case we will add a bunker and a hut.

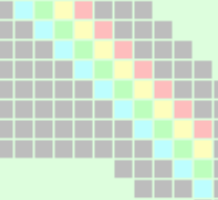
\$28 NME with rocket launcher

This takes the form \$28, \$XX where XX is a number \$01-\$FF. The easiest way to remember this is that the lower the number of XX, the greater the ferocity of the attack will be. When the player has reached a certain point and you wish to turn the attack off, simply add \$28, \$00 into your code.

Example 1

```
$28, $10 ;A nasty attack  
$00, $60 ;over this area of land space  
$28, $00 ;Attack finishes here
```

The above example is at its most simplest. You can in fact place any number of items or commands in between starting the attack and finishing it.



You can also change the frequency of the attack as you go.

Example

```
$28, $40 ;varying the attack
<perform some action>
$28, $1E
<perform some action>
$28, $00
```

\$81 Bunker

Pick a column for the bunker from \$00-\$1A. Now depending on the column you chose, the bunker will be drawn facing left or right automatically.

\$94 Big and small huts

Choose a column \$00-\$1A for the small hut. For the larger hut add \$80 onto the column. Max. Column value for large huts is \$19. Adding \$80 gives \$99.

Example

```
$94, $09 ;draws a SMALL hut in column
          $09
$94, $89 ;draws a LARGE hut in column
          $09 i.e. $09+$80
```

Now we will include this data into our program.

```
$28, $30 ;initiate the attack by the
          NME with rocket launchers
$00, $36 ;space
$94, $8B ;draws a large hut in column
          $0B
$00, $47 ;space
$81, $05 ;draw a bunker in column $05
$00, $63 ;space
$28, $00 ;end of NME with rocket
          launchers
```

Next, we will insert a road.

\$04, \$05, \$06 Roads and how to construct one



Always use the green cross code, cos I won't be there when YOU cross the road!

Roads run diagonally down from the left across half the screen.

```
$05, $10 ;from the middle of the
          screen draw a line
          diagonally down.
$00, $3F ;Space i.e. width of road.
          This value gives normal road
          width
$05, $00 ;before starting the bottom
          edging from the LHS of the
          screen
$06, $10 ;from the middle of the
          screen draw a line
          diagonally down.
$00, $3F ;Space i.e. width of road.
          This value gives normal road
          width
$06, $00 ;before starting the top
          edging from the LHS of the
          screen
```

Then we need to add road markings. Here is the finished code:

```
$05, $10 ;draw road bottom edge from
          column $10 diagonally down
          right
$00, $32 ;space
$04, $1B ;draw road marking in column
          $1B
$00, $0C ;space
$05, $00 ;draw road bottom edge from
          column $00 diagonally down
          right
$00, $0C ;space
$04, $15 ;draw road marking in column
          $15
$00, $06 ;space
$06, $10 ;draw road top edge from column
          $10 diagonally down right
$00, $12 ;space
$04, $0F ;draw road marking in column
          $0F
$00, $18 ;space
$04, $09 ;draw road marking in column
          $09
$00, $13 ;space
$06, $00 ;draw road top edge from column
          $00 diagonally down right
$00, $05 ;space
$04, $03 ;draw road marking in column
          $03
```

REACTOR

To make a road wider we can add space in between the \$05,\$00 and \$06,\$10 above i.e. \$00, \$50 gives a very wide road.

```
$05, $10 ;from the middle of the screen
draw a line diagonally down.

$00, $3F ;Space i.e. width of road.
This value gives normal road
width

$05, $00 ;before starting the bottom
edging from the LHS of the
screen

$00, $50 ;extra space to make a wider
road

$06, $10 ;from the middle of the screen
draw a line diagonally down.

$00, $3F ;Space i.e. width of road.
This value gives normal road
width

$06, $00 ;before starting the top
edging from the LHS of the
screen
```

Then it's just a case of altering the space to place the road markings correctly. The following code adds three road markings. You can play around with the code and design it just how you like.

```
$05, $10 ;draw road bottom edge from
column $10 diagonally down
right

$00, $3f ;space

$05, $00 ;draw road bottom edge from
column $00 diagonally down
right

$00, $18 ;space

$04, $1C ;draw road marking in column
$1C

$00, $2D ;space

$04, $10 ;draw road marking in column
$10

$00, $0A ;space

$06, $10 ;draw road top edge from col-
umn $10 diagonally down right

$00, $24 ;space

$04, $04 ;draw road marking in column
$04

$00, $1A ;space

$06, $00 ;draw road top edge from col-
umn $00 diagonally down right
```

Insert the code for the normal road into your code. After we have traversed the road we will insert another spawn point. At this point we will also turn on the audible alarm and have another attack of NME

with rocket launchers.

\$27 Alarm

When you would like the alarm to sound during a game you switch it on with a '1' and turn it back off after a certain distance with a '0'.

Example

```
$27, $01 ;Turn the Alarm ON
.... ;Perform some action
$27, $00 ;Turn the Alarm OFF
```

Adding this code into our program:

```
$22, $00 ;spawn point
$00, $5B ;space***
$27, $01 ;turn the alarm ON
$28, $0A ;initiate attack by NME with
rocket launchers
$00, $88 ;space (attack over this
distance)
$28, $00 ;end of NME with rocket
launchers
$27, $00 ;turn the alarm OFF
```

*** take particular note how much space I have assigned before enabling the alarm. If you were to start from the spawn point the alarm will sound again. In the original game at the end of levels 4 and 8, not enough space is assigned and although the alarm is correctly turned on, you never hear it if starting from the last spawn point due to not enough space being assigned before initialization.

You can of course add more space than that which is shown but the figure above is the absolute minimum for restarting the alarm after a spawn point. Finally, we will add the fortress:

The Fortress

At the end of every level we need to build the

```
$09, $0B ;draw wall from right to
left starting in column $0B
$09, $1F ;draw wall from right to
left starting in column $1F
$21, $14 ;draw the fortress gates in
column $14
$00, $16 ;space
$22, $01 ;signal end of level and
open the gates
```

fortress. This is achieved with the following code

This code will always remain the same except in cases

where we want additional soldiers to be firing from the top of the wall. In this case we change the \$00, \$16 above to \$00, \$1E.

```
$E323    LD A, ($FDE5)    ;level
$E326    AND 03
$E328    JR NZ           ;no soldier on
                        end gate
$E32A                    ;soldiers on
                        gate routine
```

In the game code itself starting at \$E323 resides the routine for the additional soldiers on the wall.

In the original game the AND 03 will make the gate top soldiers appear on levels 4 and 8. If we wanted the gate top soldiers to ALWAYS appear then we would never make the jump at \$E328. This is achieved with Poke \$E329,0 (Poke 58153,0). For the gate top soldiers NEVER to appear you would Poke \$E328,\$18 (Poke 58152,24).

You can also change the value next to the AND above for some more variation.

This can be achieved with Poke \$E327,X (Poke 58151,X) where X can be for example:

```
01 ;gate top soldiers will appear on levels 2,4,6,8
02 ;gate top soldiers will appear on levels 1,4,5,8
03 ;gate top soldiers will appear on levels 4,8
04 ;gate top soldiers will appear on levels 1,2,3,8
05 ;gate top soldiers will appear on levels 2,8
```

Just make sure you remember that if you code the level to have gate top soldiers then the space before the end of level signal must be \$00, \$1E and not \$00, \$16. If there are too many NME soldiers already on the screen when the fortress is reached, gate top soldiers will never appear anyway.

Making your own copy of Commando

After you have built your own levels it is time to finalise your 'personal' copy of Commando. You will need a TAPE image (TAP, TZX) of any copy of Commando.

Snapshots of any kind (Z80, SNA) are **NOT** usually acceptable. However, where a snapshot has been taken before the game has had chance to run even just one byte, such as the 128K version with added music, then this is permitted.

For emulators with an assembler (Spin, EmuZWin)

1. Open the debugger and set a breakpoint at \$641E (25630).
2. Load your copy of Commando.

3. When the program breaks, switch to the assembler and assemble your new data.
4. Remove the breakpoint and exit the debugger to return to the game.
5. Save a snapshot.
6. Test thoroughly.

If your version of Spin is older than V0.6 then use the following method:

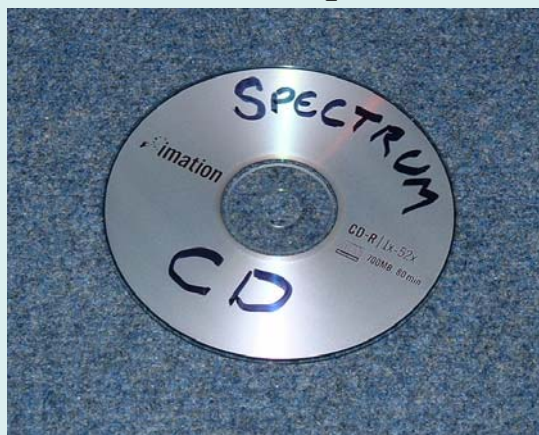
1. Open the debugger and set a breakpoint at \$641E (25630).
2. Load your copy of Commando.
3. When the program breaks, go to the EDIT menu and select 'Edit Mode'.
4. Enter JP \$641E in place of JP \$69D2.
5. Remove the breakpoint and exit the debugger.
6. You are now able to use the assembler.
7. After assembling return to the debugger, select Edit Mode again, this time returning the original instruction of JP \$69D2 (27090).
8. Exit the debugger.
9. Save and Test.

For emulators that can load a binary file (SpecEmu, Spectaculator)

This method is slightly more complicated. You can use notepad or such like to build your levels. When you are happy with your work, save the result as **filename.ASM**

1. Download Pasmu or other assembler and create a binary file of the new levels. Take note that you are creating a binary file of the levels only with this method. Thus your ORG will be \$9236 (start of level one). No other ORG must be used. See step 6 below to add further pokes. For Pasmu the command is **Pasmu filename.asm filename.bin**
2. Open Spectaculator and set a breakpoint at \$641E (25630).
3. When the program breaks you will see \$641E C3 D2 69 JP \$69D2
Click on the C3 D2 69 and change to \$C3 \$1E \$64
4. Remove the breakpoint and close the debugger.
5. Select 'Open File' and load the filename.BIN that you created with Pasmu.
6. Use the 'Poke Memory' option from the tools menu to add any extra pokes you require i.e.
Poke 56567,X ;X=your character start column
Poke 56570,Y ;Y=how far down the screen your character starts.
Poke 58153,0 ;gate top soldiers on every level
7. Open the debugger once more and navigate to \$641E
Click on the C3 1E 64 and change back to \$C3 \$D2 \$69
8. Exit the debugger.
9. Save a snapshot.

...Play CD's on your Speccy!



It's a question that gets asked over and over again... How do I save my games to CD & play them on my machine? Well, Graham Richards has the answer.

More and more tapes in our collections are degrading, and games that used to load first go are now sounding a little tired, especially if they've been stored in the loft for 20 years! For those of us who are not so keen on emulation, and who prefer real-media gaming on original machines, the best way to preserve our games is to record them onto audio CDs. These can be easily backed up, time and again, long after the tapes have turned to dust. (Sounds like some scare-mongering advertisement!)

For this, you will need:

- A ZX Spectrum (48k,+,or +128k)
- A selection of Speccy games on cassette
- A PC or Laptop with soundcard inputs or mic socket
- A tape recorder (or spectrum 128+2, +2a)
- Mains powered speakers with headphone socket
- A CD player
- Audiolab cleaning lab 1.0 or better
- Steinberg WavelabLite 1.3 or better

Before we start burning off our entire Spectrum games collection to CD, we're going to have to turn them into WAV files. For this we need to connect a tape recorder (Or 128+2) to our PC.

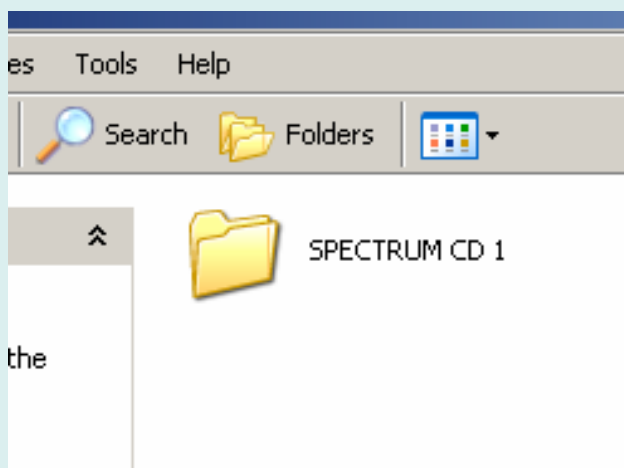
This can be linked to either the soundcard line-in socket, or if you don't have one of these, then the microphone socket will do the job just fine. (Looks weird doesn't it!)

You're also going to need some software to manage your file, and for this, I'd recommend 'Magix Audio Cleaning Lab Deluxe' or 'Audiolab' for short, as it has an easy to understand interface, and can manage many files. You can however use any audio software that exports uncompressed WAV files.



STAGE ONE:

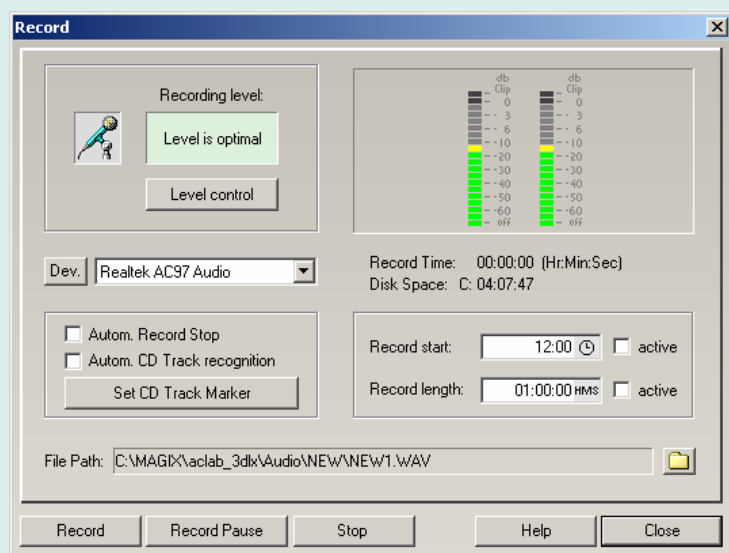
Firstly, create a new folder on your PC, where your WAVs will be saved. I've called mine, 'SPECTRUM CD'



Now it's time to load in Audiolab and record your first Speccy game. Select the 'Record Audio' option, and then choose your method of audio entry from the

'Level Control' dialogue tab. All that's left at this moment is to queue-up your tape recorder, press the Play button, then hit, 'Record.'

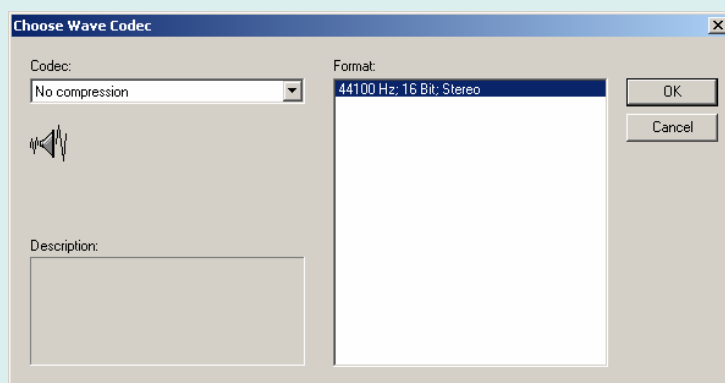
On Audiolab, remember that the best results are achieved when the volume bar is just in the yellow.



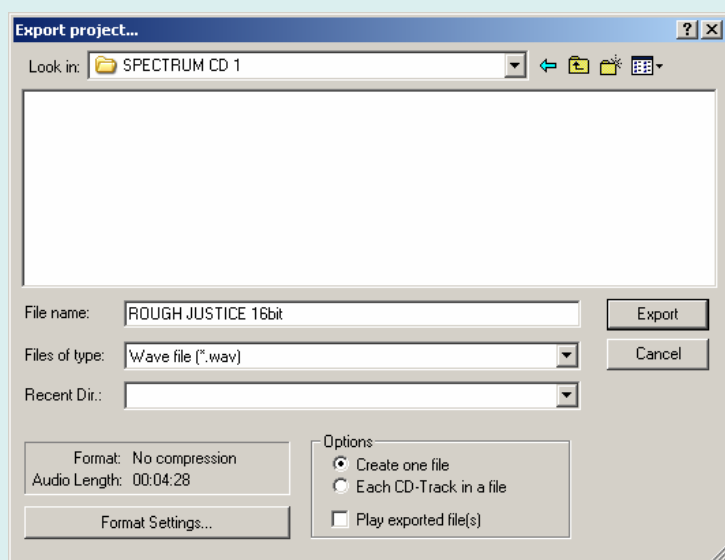
Adjust the volume on the tape recorder, to achieve this. (Cor, those were the days!) Also remember to de-select the Autom. Record Stop and Autom. CD Track recognition. You'll find parts of your program disappearing if you don't!

If using other software, try the tape recorder at half volume or use a 128+2 so that the volume is pre-set.

Once the Audio is recorded, it can be trimmed if necessary, and then exported to your chosen folder.



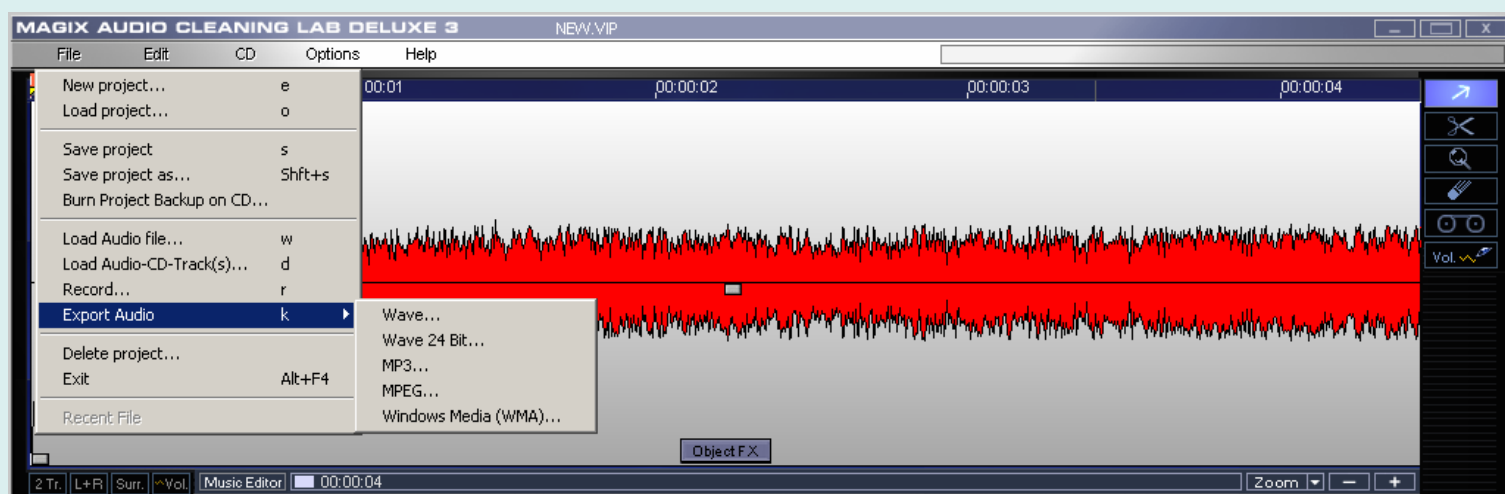
Here, I've chosen, 'Rough Justice' by Jonathan Cauldwell. At this point, it is a good idea to specify in your titling, that the file is still 16bit.

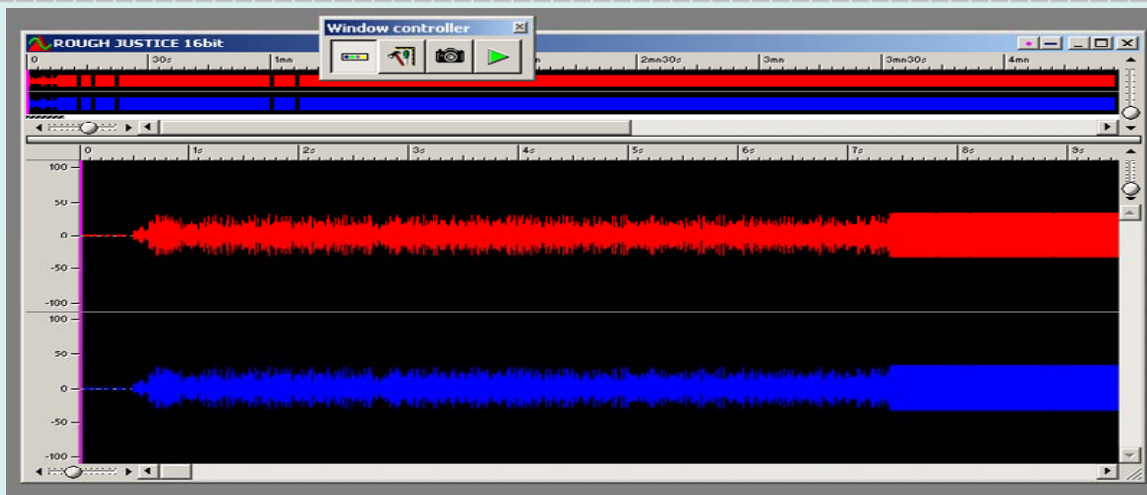


STAGE TWO:

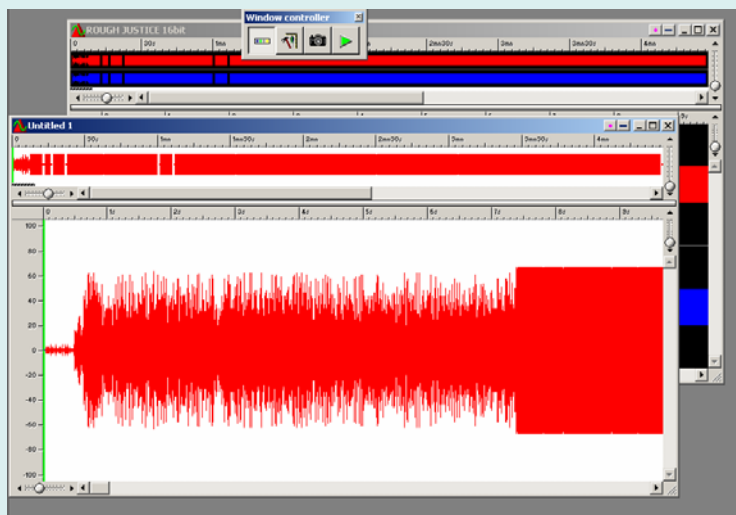
So, we've saved it, but we can't play it just yet, as the result is (If we've been using Audiolab) a 16bit stereo file, and this just won't do for what we want...We need to do something about it!

Load in WavLab and open the game file you just created. You will see the game's soundwave in stereo.





To Remedy this, go to the, 'File/Edit Audio Attributes' dialogue and change the Bit Resolution to 8bit. Press Ctrl+A to select the sound wave and drag it off the screen. A stereo icon will be displayed, but will change to mono, as soon as you hit CTRL again. Release the mouse button, and a new Mono file will be created!



Save this new file as 'Name8bit.WAV' or something like that, just to distinguish it from your previous save. You can now delete the 16bit version of the file.

Hey, you can play this 8bit file in most emulators...But for us classical gamers, it's not over yet.

Repeat stages One and Two, until you have about 10 of your favourite games, ready to go!

STAGE THREE:

It's no good just burning this lot to a CD as it stands, as Speccy's just don't play WAVs, and neither do most cheapo CD players, so we're going to have to burn them as Audio CDs. Media player will suffice for this, or even Easy CD creator, but try to avoid, 'Record now', as it has the nasty habit of adding 'noise' to your final burn. Again, I'll be using Audiolab, as it has a rather convenient, 'Export Audio CD' button, and it kindly adds up the minutes of your WAVs, so you don't overrun on time.

(70mins is about usual).

Firstly, load all your 8bit Wav files into the software, and click on, 'Burn CD'. All the games will be saved within their own CD track. You may want to write the titles down on the CD case, so you know what's on it... (Hey this is just like when I used to pirate games for my mates at scho...Erm, heh, not really!)

Before you sit back and let your PC do its thing, you may find it a good idea to change the speed of the recording to x1, as some stand-alone CD players don't like quickly burnt discs. Now make a cup of tea, or read through the WOS forums while it churns away to itself.

STAGE FOUR:

Nearly there. Now, connect your CD player to the speakers, and connect the speakers to the Speccy via the headphone and ear sockets.



Turn the speaker volume up to Max (or just below) Type Load "" and select a track. Bingo! (Or Agent X, or whatever.)

Your games should load first go every time, so long as the original tapes were in good order! Have fun with it, folks.

The WOS Preservation
Team Presents
The Grand Opening of
**The Preservation
Team Shop**

www.zxsoftware.co.uk/zxszen

Orders will be taken from midnight on 20th Aug 2006

Hundreds of Spectrum titles available.

**Any money raised will be going towards the
WOS preservation projects.**

Sensible prices and no rip offs for postage.

**Help us to help you, check out the shop
before you bid at auction..**

Donations of tapes etc gratefully accepted.

For more details contact adjb@zxsoftware.co.uk

FEATURE



Dumba de dum... dumba de dum... dumba de dum...
diddle dum...

WAH-WAH - AAAAAAAAAAAAAAAAAHHH!!!

In case you've had your head stuck face down in the cat tray for the last year or so, you couldn't have failed to notice that Doctor Who is BACK! Not only is he back, but he's back on the right station, at the right time (well, Saturday evenings anyhow), he has his old faithful TARDIS, still a Police Box, his sonic screwdriver.. AND to top all that, it's bloody good, on-the-edge-of-your-seat, hide-behind-the-sofa good old fashioned British sci-fi!

After a gap of sixteen years or so, a whole new generation have finally got their own Doctor! Although the git has already regenerated into Doctor Number ten (What the??) Still, that's all in the happy happy here and now! Doctor Who is back and here to stay... Well, for a while anyway.

Many people of my age will remember what a BIG thing science fiction was in the late seventies and eighties. Star Wars was the defining moment in a lot of our lives and the mass tidal wave of science fiction that spewed out of Hollywood was inevitable.

We had Battlestar Galactica, Buck Rogers in the 25th Century, and Captain Zep - Space Detective... Whoops... that one was ashamedly ours... But this 'all new' wave of glossy, special effects driven Sci-Fi couldn't match our very own good Doctor, who was getting ready to celebrate his 21st birthday in the very early eighties. He was established, tried, tested and proven. Naturally, when the Sinclair ZX Spectrum began to mysteriously materialize in homes all across the UK; it wouldn't be long before the Doctor would follow.

Doctor Who Genesis of the Spectrum

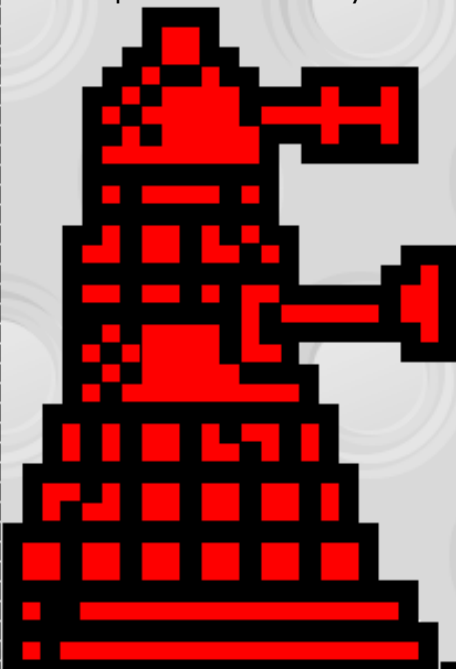
All across the United Kingdom, small black slabs with rubber protuberances are appearing in homes, they attach themselves to televisions sets, hypnotising youngsters everywhere. When parents attempt to break the connection, children, eyes glazed and vacant turn and announce in monotone voices 'But - It - Will - Help - Me - With - My - HOME-WORRRK!'

THE KEY TO TIME

Lumpsoft - Published Through Sentient Software
Released in 1984

Original Selling Price £5.95

Available at World Of Spectrum



```
I am in the Tardis control room.  
I can also see:-  
Tardis control console.  
Closed outer doors.  
Arched doorway.  
Small spherical crystal.  
Tardis View Screen  
  
Tell me what to do.  
>
```


The Key to Time was NOT based on the seventeenth season of Doctor Who stories known by same name. You took control of the Doctor, starting off in his first incarnation (William Hartnell, who as a point of interest, the game was dedicated) each time you died, you would regenerate up the line until you became Colin Baker (sixth, and up-coming Doctor at that time.) The game itself is a typical text adventure, no graphics but a heavy and useful help feature made it easier to get into than other games of this ilk.



The first game to materialise on the ZX Spectrum based on the Doctor's exploits was 'The Key to Time' this was, in fact, the third Doctor Who game to appear for home computers. The first being a type-in from the March 83' issue of 'Computer and Video Games' for the Atari 400 / 800. The first 'Officially Licensed' computer game was 'Doctor Who - The First Adventure' surprisingly for the BBC Micro that was seen heavily in most secondary schools in those years.

An early and impressive QUILL written adventure game, extremely addictive for the Doctor Who fans of the day that was lucky enough to get their hands on it. It's interesting to point out that it was also only released on the ZX Spectrum too. No conversions to other platforms followed.

Released by Lumpsoft, your typical cottage-industry software house, it was available mail order only and failed to grab much attention. Reviews appeared in Crash and Sinclair User, but no mention was seen in 'Doctor Who Monthly' The ideal place to advertise the game... or so you would think... Might have something to do with the fact that no licence was applied for, hence no 'direct' reference to 'Doctor Who' even though Daleks, Cybermen and Sensorites were all used in the different levels.

Crash reviewed the game, giving it a score of seven (out of ten) a fair score for a well written and lovingly polished text based adventure.

All in all, a good first outing for the Doctor on the Speccy.

DOCTOR WHO AND THE WARLORD

BBC Software

Released in 1984

Original Selling Price N/A

Missing In Action

DOCTOR WHO AND THE WARLORD

The year is 2743. You arrive with the Doctor on the planet Quantain, where a century-long war is just coming to a close.

You emerge from the TARDIS to find deep mist all around. Travelling North you stumble into one of the last battles of the long war. The Doctor, However, seems convinced that King Varangar, (an old friend of his) is in some deadly danger, and presses on.

PRESS ANY KEY

The second adventure game featuring the good Doctor from the BBC, and a near miss for the Spectrum. The game itself was a simple enough, much like 'The First Adventure' although the BBC had brainstormed after the poor sales of the latter, and gone with a neon TARDIS on the cover, rather than a rendition of the Time Lord of the day, sadly, and what soon was to become a common story, the game failed to be released on the ZX Spectrum, even though it had been written.

THE MIND MAZE

Kevin O'Shea

Written in 1985

Original Selling Price N/A

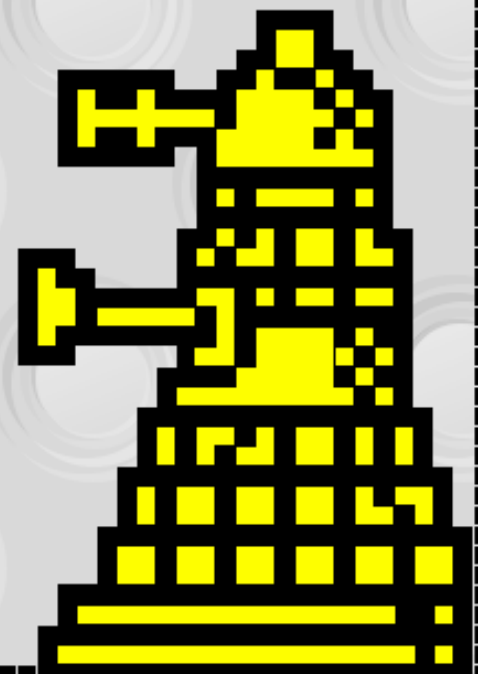
Available From Kevin's Website, SHOCKINGLY not on WoS!

The Mind Maze was the first of two games written by

Doctor Who fanboy Kevin O'Shea, and BOY does this guy know his stuff. The game (Yup - another text adventure) finds the Doctor in his Sixth incarnation, battling his way through the maze,

the prize, a circuit from the TARDIS. The Maze, an attack on his psyche, making memories of enemies past solid, and out to get him.

It is important to





note that this game was never meant for public release, it only surfaced when Kevin placed it on his website as a piece of fun, and soon found it spreading throughout the internet, on Doctor Who fan sites, and Spectrum sites alike.

It's a bit of a shame, as, for a Doctor Who fan, this game would have been the MUTS back in 85' absolutely DRIPPING with Whovian titbits and references, it would have kept many a fanboy happy through those long winter nights under the anorak.

DOCTOR WHO AND THE MINES OF TERROR

Micropower / BBC

Released in 1985

Planned Selling Price £14.95

Missing In Action



Are YOU ready for brain to brain combat?

We had strange adverts, huge amount of hype, excessive prizes in Crash, including a trip to the Doctor Who studios and endless promises that the game was just around the corner.

Doctor Who and The Mines Of Terror! The first 'Official' Doctor Who game to be written for the ZX Spectrum.

Did it materialise?

Did it buggery.

It was talked about for months. All the key spectrum magazines had their say, it's out next month, it's been held over until next year, and we think the price is too high for spectrum owners so were going to release it at £11.95 instead of £14.99.

The wait went on. The rumours dried up and eventually everyone accepted that 'Doctor Who and the Mines of Terror' was never coming.

At the last minute it was shelved. Why? Who knows? Maybe it was the poor reviews and sales of the BBC and Amstrad versions of the game, maybe it really sucked. We will never know for sure. But we do know it was finished, so one day it may well materialise, just several decades too late.

It would have been a platform arcade game featuring the sixth Doctor and a robot cat called Splinx (Come back K9 all is forgiven!) that was also rumoured to be making an appearance in the series, which, of course, it didn't.



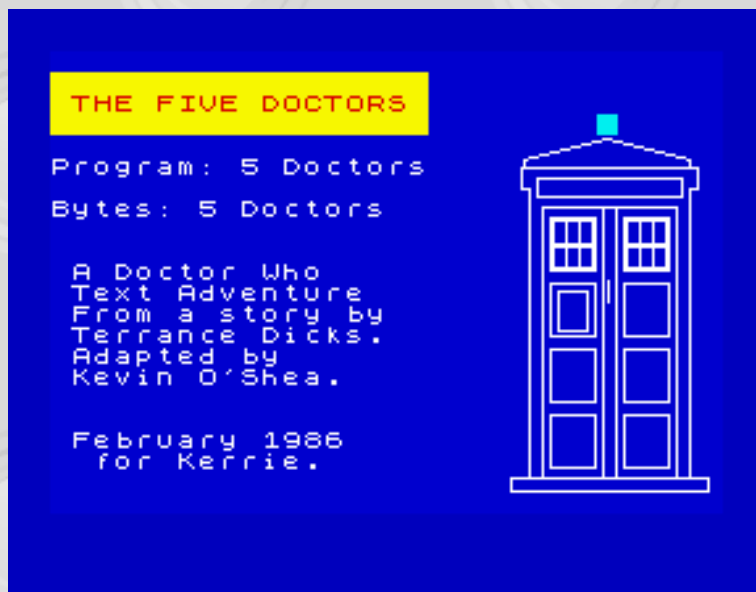
THE FIVE DOCTORS

Kevin O'Shea

Written in 1986

Original Selling Price N/A

Available at World Of Spectrum



Kevin's back with his final Doctor Who game for the Spectrum. This time its 'The Five Doctors' based closely on the 21st Anniversary story of the same name. It is also important to note that this game, like its predecessor, was not written for commercial release, just for the love of the program.

The game itself is a lot more polished than 'The Mind Maze' and eases back on the nostalgia overload, although this story was in any case a massive stroll down memory lane! If you have seen the feature length episode, or read the book by Terrance Dicks, a adventurer would find this game a walk in the park. But that's not the point, the point is it's damn good fun and again, a damn shame it was never commercially released.

DOCTOR WHAT!

CRL Group

Written in 1986

Original Selling Price £7.95

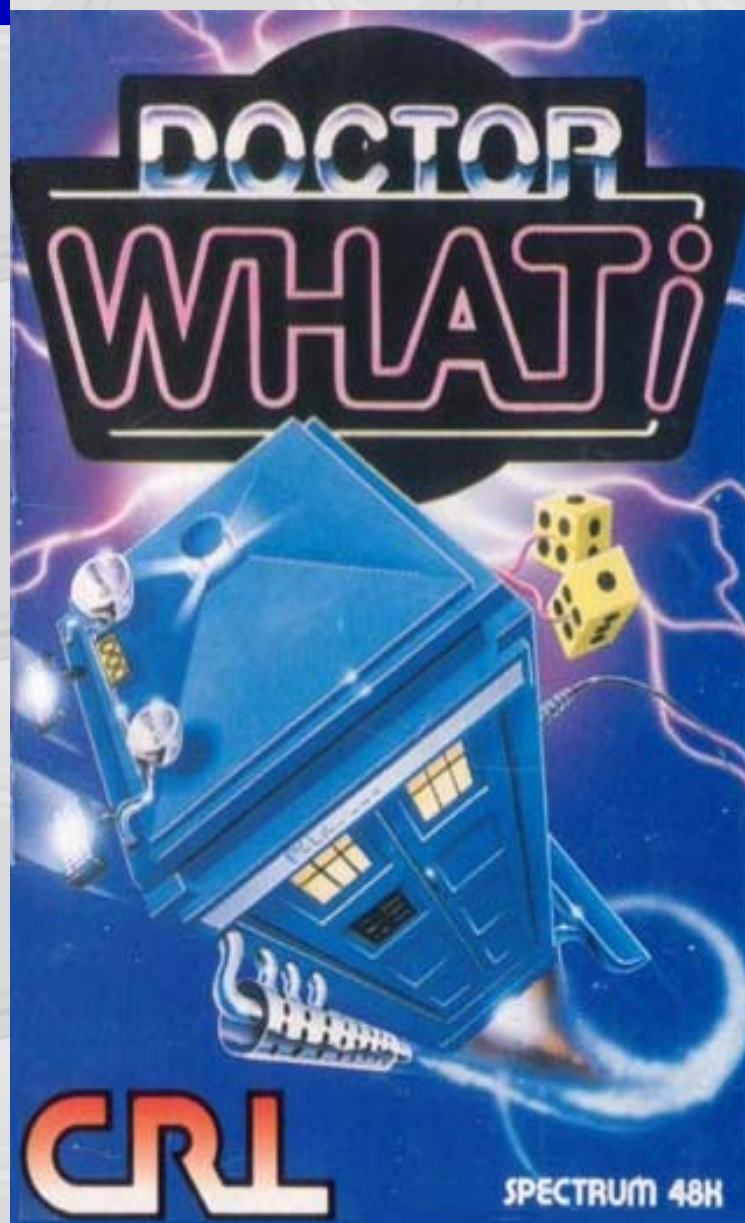
Available at World Of Spectrum

The next game to feature our beloved time lord was a cheeky attempt at pushing the boundaries of copyright infringement.

Doctor What! was an out and out parody of the series, released in 1986, it featured four 'Doctor's' – Doctor What, Where, When and Why (no Who though) laced with a pimped Police Box, sonic screwdriver, jelly babies and Daleks, this game was the first real arcade game taste Doctor Who fans got.... And in all honesty, all it really did was rub salt into the wounds.

It was not well received. Non-existent sound bar a few ropery spot effects, bad use of colour and game play

from 1982 sealed the games fate. Crash gave it an abysmal 17%, Sinclair User gave it a poor 2 out of 5 and Your Sinclair, well, they were feeling generous and gave it 6 out of 10.



It was a tired formula for a game, similar to Finders Keepers, but a long time after and a lot more expensive. Still... The Cover was nice.

DOCTOR GOO AND THE SAMAROONS

Global Games

Written in 1991

Original Selling Price £

Available at World Of Spectrum

Then things went quiet.

Doctor Goo and the Samaroons surfaced in 1991...
Another adventure game. Another Parody. Another
Disappointment.

```
This is Dr Goo's time machine.  
It is quite bare apart from a  
large red button set in one of  
the walls.  
Exit:Out.  
Dr Goo can see:-  
A letter  
What now?  
>
```

The spectrum moved on, new computers came out;
the home videogame console market had its second
birth.

Then, in 1992, when the spectrum was in its twilight
years, rumours of the Doctor's return began again.

Admiral software had bought the rights to the series
and was planning a massive cross-platform campaign,
Dalek Attack was to be release on no less than six
formats, including the Spectrum.

colourful Commodore Amiga Graphics, with a small
footnote reminding people that you wouldn't get that
out of a lesser machine.

Again, we waited, held our breath. Coverage in Your
Sinclair, now the only game in town was extensive, it
was on its way. It was really going to happen, then, of
course, the rumours began. It was going to be
shelved. There was no life left in the Spectrum
Market.

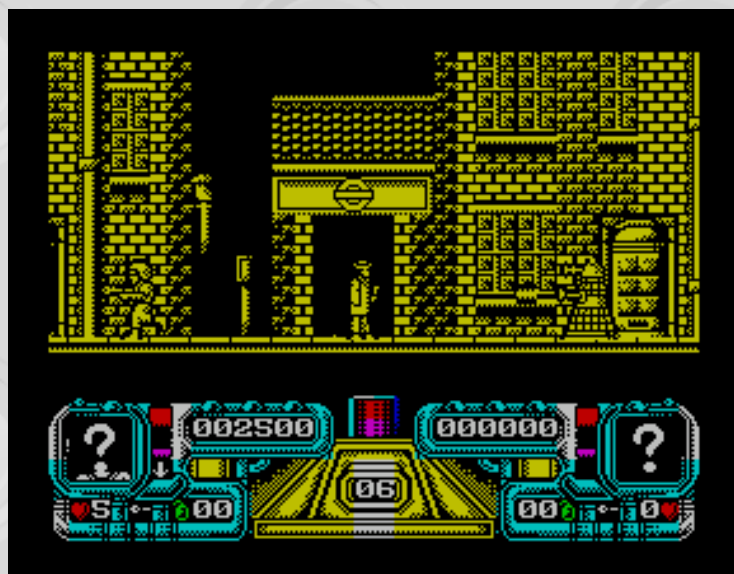
As it turned out, it was the Amstrad version that was
shelved. Dalek Attack was released with an
exceptionally reasonable £6.99 price tag. The Doctor
in all his glory had finally arrived on the ZX Spectrum.

Compared to the 16bit versions, the game was
extensively watered down. Smaller levels, no multi-
Doctor player Choice, bad rendition of the Doctor Who
theme. How could it live up to the hype and
anticipation?

Short answer - It couldn't. After all the hype, Your
Sinclair was far from generous with their review.

Complaining about the two player mode, occasional
doorway bugs that could hardly be described as bugs
and 'shabby design'

Dalek Attack received a measly 56%



It's quite sad really as 'Dalek Attack' is a very
competent platform game. When researching this
article I spent a lot of time playing the game on-line
and found it highly enjoyable.

You start off in a sewer to finally emerge in London,
then battle your way through Paris, America, Tokyo
until you finally arrive on Skaro, the Daleks Home
Planet and a final confrontation with Davros no less.
The levels are designed really well, the animation is
pleasing. It was the timing that was wrong.

If Dalek Attack had been released four years earlier it
may well have been a mega game. Sadly it was too

Adverts came out showing off the wonderfully

DALEK ATTACK
COPYRIGHT BBC 1973-1992
LICENSED BY BBC ENTERPRISES LTD
DR WHO AND THE DR WHO LOGO ARE
TRADEMARKS OF THE BRITISH
BROADCASTING CORPORATION AND
ARE USED UNDER LICENCE

PRESS SPACE TO START

T - TOGGLE ONE/TWO PLAYER GAME

(C) 1992 ALTERNATIVE SOFTWARE



little too late. It is also important to note the time the game was released. It truly was one of the final mass release titles on the Sinclair ZX Spectrum.

How could it, how could anything on the Spectrum compete with the likes of Sonic the wizz popping junkie hedgehog. Well.... That's what people wanted in 1992.... Well, some people at any rate.

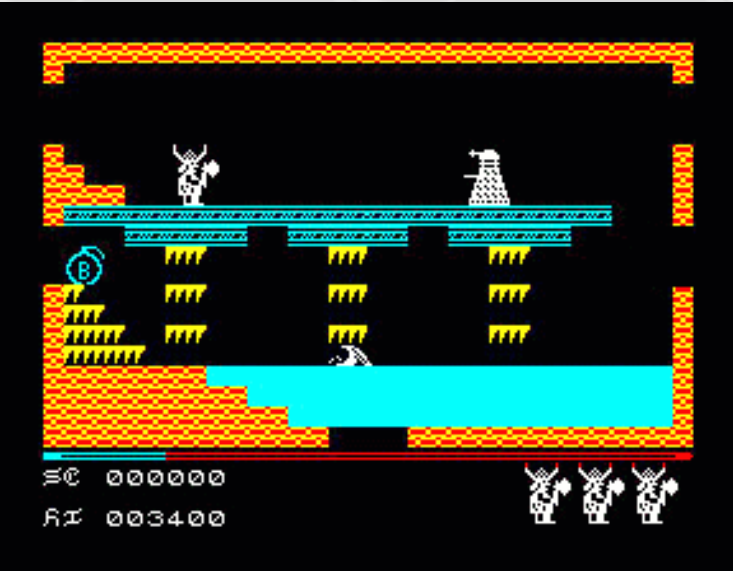
The Doctor had finally arrived on the ZX Spectrum. But true to his television counterpart, he had arrived several years too late.



Dalek Spotting

Although games featuring the BBC's favourite Time Lord were few and far between, his number one enemy never failed to turn up unannounced in various shapes and forms throughout Spectrum game History. Here's a few -

BRIAN BLOODAXE EDGE

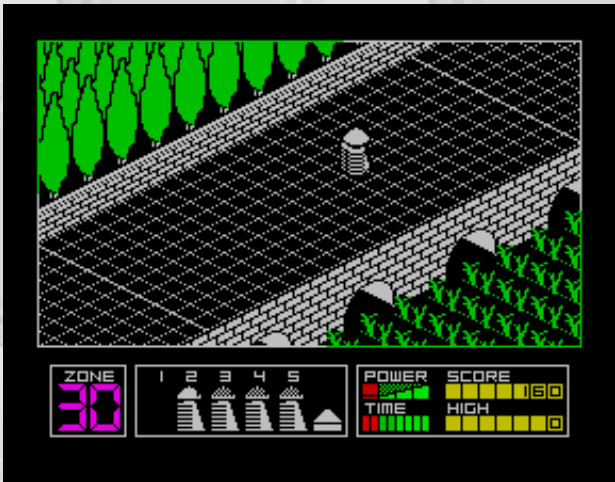


No question with this one. Not too far from the start of this Monty Python tribute game we find a Dalek, and not just any Dalek, a very well designed and animated one at that!

HERBERTS DUMMY RUN MIKROGEN



AH! The Mini Game! In the depths of Herbert's Dummy Run just like in Pjamarama, we find an invaders style mini game... SO... What do those invaders look like then??



HIGHWAY ENCOUNTER / ALIEN HIGHWAY VORTEX SOFTWARE

It's mentioned in many places that the pepper pot from Highway Encounter was meant to be a Dalek. Me, I never saw it. You can decide for yourselves.

Well. That about wraps it up. We still live in hope that one day those missing written adventures will surface and take their place in the grand archive of all that is spectrum. Until then, you never know, Kevin O'Shea may still dust down his Speccy and start writing adventures again, maybe even based on the Doctors New Adventures.

Time will Tell. It always does.

...Create a text adventure using PAW

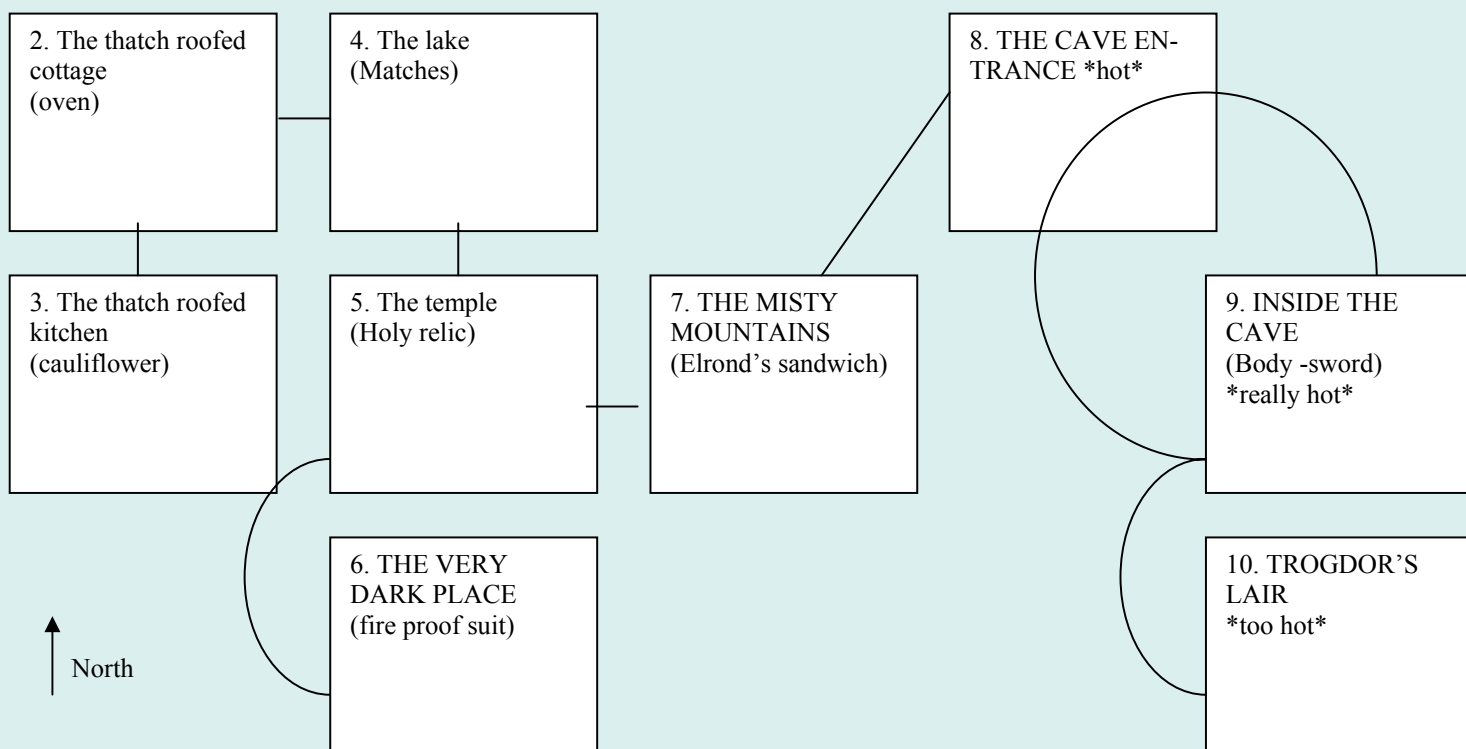
Who among us hasn't experienced the thrill of OPENing the round green DOOR and going EAST? Not to mention SAYing to ELROND "GIVE ME LUNCH". And of course there's my personal favourite GET TREASURE. I'm talking of course of adventure games. And if liberating treasure in the Hobbit isn't your thing then what about avoiding the Mandroids in the Necris Dome? Or maybe drinking some ginger beer in the Famous Five?

It's all good fun but while playing, some of us have thought to ourselves, "I wish I could make a game like this." Well, you can, if you have an adventure writing utility like Gilsoft's Professional Adventure Writing System (PAW). There are others, for example Incentive Software's Graphic Adventure Creator but this tutorial is only going to be dealing with PAW. I'm not going to try and cover everything all in one go. This tutorial will cover the basics in a suitable order and by the end of the last part we'll have made simple adventure game and you'll have learned enough techniques to go and write your own.

Most text adventures allow the player to input commands in a VERB NOUN format. However, games created with PAW allowed the player to input much more complex sentences; for example, it is possible to QUICKLY GET THE SHORT SWORD AND KILL THE UGLY GOBLIN or even GET THE SWORD AND THE SANDWICH. PUT ALL INSIDE THE BAG. PAW also allows the player to talk to different

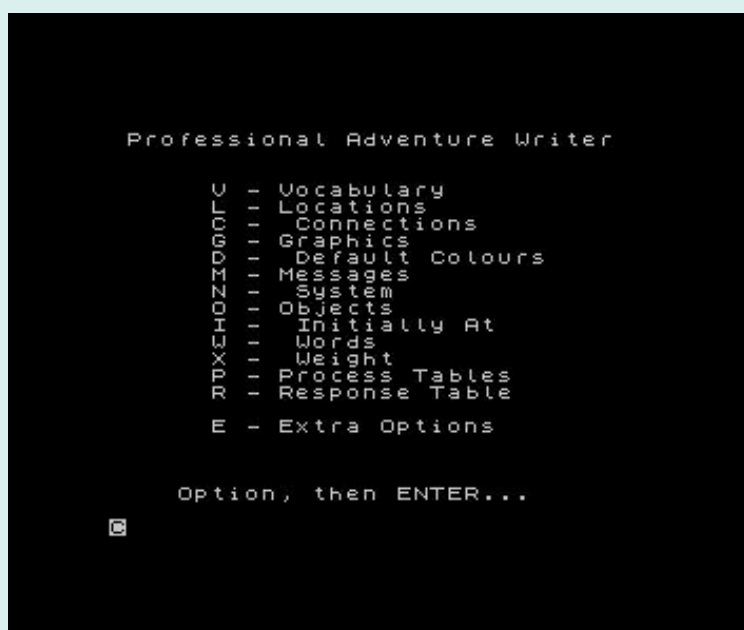
characters. Other characters breathe a little life into the adventure and can be made to appear to have a life of their own. Such characters are known as Pseudo Intelligent Characters. It's possible for a PSI to move around of their own accord, take or drop items, and often have some use in the game. Perhaps they can perform a task that the player cannot. So the player could type SAY TO BEN "GIVE TREASURE TO ME" but obviously Ben would not. So the player would have to find some way of getting Ben to part with his treasure.

Before we get our teeth stuck into PAW it's worth saying that writing adventure games is very difficult if you don't plan what you want to do in advance. Personally, I like to draw a map first. Locations are represented by boxes with a brief description in them and are connected with lines to indicate where the player can go from each location. Curved lines indicate whether the player can go up or down.



The idea of my simple (and silly) game is to try and kill the dragon. There is a sword that the player could use, or they could try and poison it with some cauliflower cheese.

Well, it's time to load up your copy of PAW (make sure you're in 128K mode) and take a look. Once it has loaded, press a key to go to the main menu. Have a quick look at what's available, then press E and ENTER to have a look at the other page. When you're ready Press E and ENTER to get first menu. It doesn't matter which page you're on, you can select any of the options by pressing the corresponding key followed by ENTER. Stay on the page with V - Vocabulary at the top for now. We won't be using the other options for a while.



The first option we're going to take a look at is Locations. Press L then Enter and you'll be taken to a new menu. Location 0 is already inserted and we need to amend it as it's going to be our title page. So type A 0 and press Enter. A quick way to delete the text that's already there just tap Edit twice. Now type:

Dragon Adventure

A terrible dragon has moved into a nearby cave and I'm afraid that my luxurious thatched roofed cottage/hovel will be burnt to the ground. I must destroy the dragon in its lair before it's too late.

And then press Enter and once again to return to the menu (to insert a new line you need to press Extended mode, 7, then delete). Now we're going to insert a new location. So press I followed by Enter. The description for Location 1 is:

I am inside the bag.

Once you've done that, insert the following locations

Location 2

I am inside my luxurious thatched roofed cottage. In the corner is a sack filled with straw and there is a box of junk pushed up against one wall. To the south is the thatched roofed kitchen. I can also go east.

Location 3

I am in my kitchen. Being a poor peasant I rarely have any food to cook in here. Nevertheless I can see an ancient oven. I can go north.

Location 4

I am on the shore of a beautiful blue lake. In the distance I can see the peaks of the mountains. I can go west and back to my cottage or I can go south towards a ruined temple.

Location 5

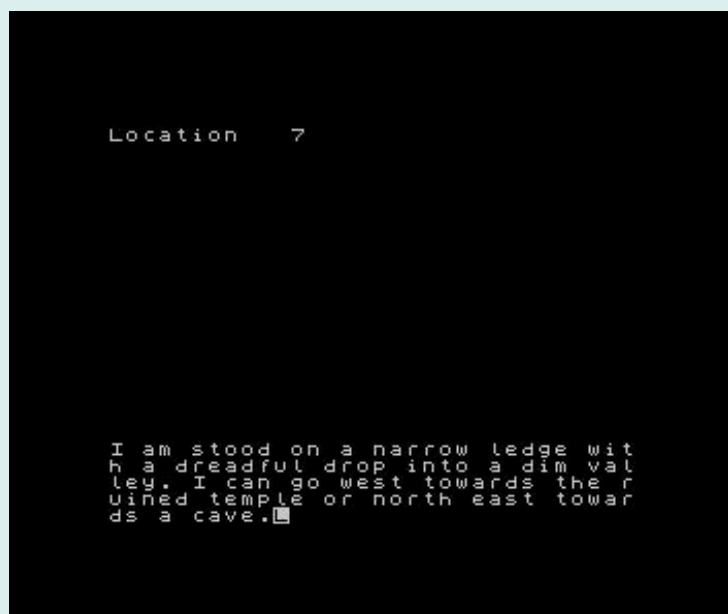
I am in a ruined temple. Most of the walls have fallen down and there doesn't seem to be much here among the rubble. Steps lead down or I can go east towards the mountains.

Location 6

I am in a small chamber beneath the ruined temple. Thieves have long since stolen all of the valuable items from here. Now all that remains is dust and a few worthless items.

Location 7

I am stood on a narrow ledge with a dreadful drop into a dim valley. I can go west towards the ruined temple or north east towards a cave.



```
Location 7
```

```
I am stood on a narrow ledge with a dreadful drop into a dim valley. I can go west towards the ruined temple or north east towards a cave.
```

Location 8

I am at the entrance to a large cave. I can go south west and back to the mountains or down into the cave.

Location 9

I am inside the cave. I can see the body of a dead adventurer and the floor is littered with bones. I can go up or down.

Location 10

I am in the dragon's lair. The place is filled with wondrous treasures. I can go up.

```
Location 6
I am in a dark chamber beneath the ruined temple. Cobwebs hang from the ceiling. Thieves have plundered this place and all that remains is worthless junk.
```

```
Location 7
I am stood on a narrow ledge with a dreadful drop into a dim valley. I can go west towards the ruined temple or north east towards a cave.
```

```
Location 8
I am at the entrance to a large cave. I can go south west and back to the mountains or down into the cave.
```

```
Location 9
I am inside the cave. I can see the body of a dead adventurer and the floor is littered with bones. I can go up or down.
```

```
More...
```

all the messages so far on the screen. You can type P followed by a valid location number and PAW will display all locations starting from that number. The print command is available in nearly all of the menus.

Z and Enter to return to the main menu. Next we need insert the connections so that the player can move around. Choose option C and you'll be taken to the connections menu.

Here you can only amend entries that already exist. Locations 0 and 1 are both special locations and don't have any connections. Type A 2 and then press Enter to amend the connections from location 2. A quick look at the map will show that you can go south to the kitchen and east to the lake. So type: S 3 E 4 and press Enter. Now do the same for each location.

Location 3: N 2

Location 4: W 2 S 5

Location 5: N 4 D 6 E 7

Location 6: U 5

Location 7: W 5 NE 8

Location 8: SW 7 D 9

Location 9: U 8 D 10

Location 10: U 9

```
Location 5      S      TO      5
                N      TO      4
                D      TO      6
                E      TO      7

Location 6      U      TO      5

Location 7      W      TO      5
                NE     TO      8

Location 8      SW     TO      7
                D      TO      9

Location 9      U      TO      8
                D      TO     10

Location 10     U      TO      9

Location 11
```

```
Press any key to continue.
```

You could have a connection from Location 0 (the title screen) to location 3 but that wouldn't look very good, to get from the title screen the player would have to type NORTH or whatever direction you chose. Instead we're going to have a quick look at the process tables. Select option P from the main menu.

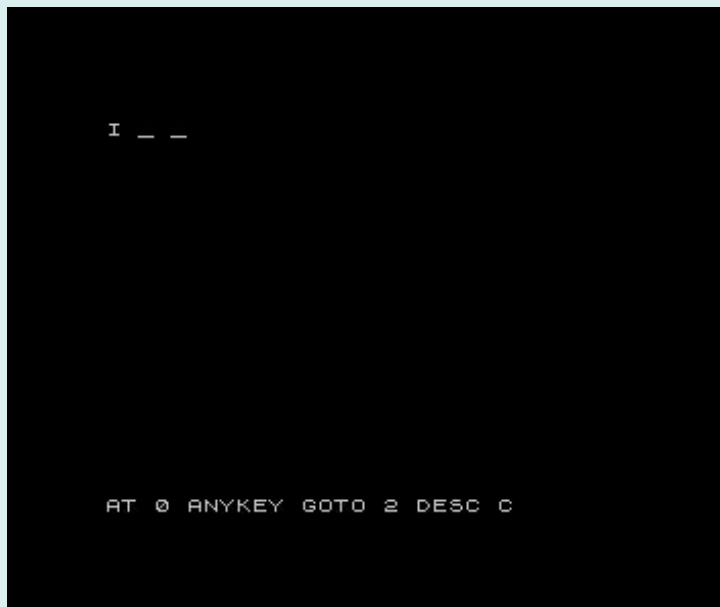
By default PAW selects process table 2 for you. This is the table that gets scanned after the player has made an input. Type S 1 to select process table 1.

Once you've added all the locations press P to print

This table is called prior to the player being allowed to type a command.

Type I _ _ and press enter. Now type the following instructions:

```
AT 0 ANYKEY GOTO 2 DESC [Enter]
```



```
I _ _  
  
AT 0 ANYKEY GOTO 2 DESC C
```

This means that when the player is AT location 0, the game should wait for any key to be pressed, then go to location 2 and describe it.

Press Z and then T to test the adventure. You'll be asked if you want diagnostics. Say no for now and just test the game. When you're bored with wandering around type QUIT and then say YES followed by NO. You'll then be brought back to the main menu.

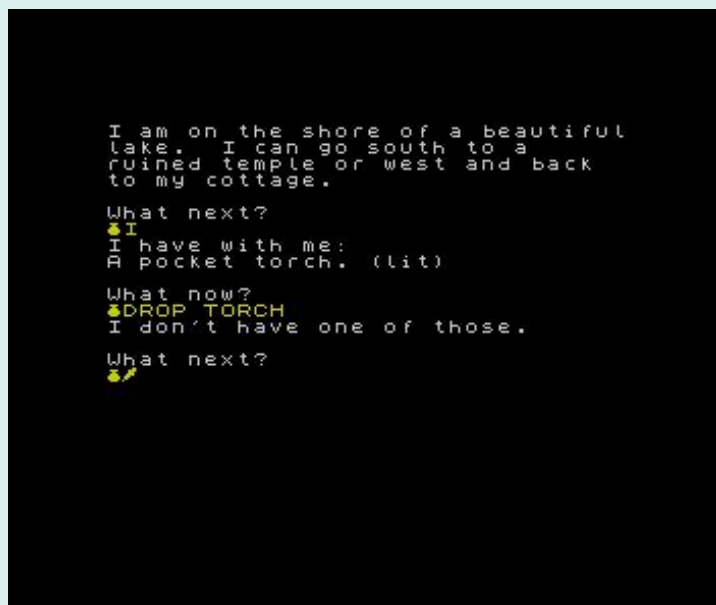
Now we're going to add a few objects. Press O on the main menu and insert the following objects:

- Object 1: A bag
- Object 2: A book.
- Object 3: A cauliflower.
- Object 4: Elrond's sandwich.
- Object 5: Some cheese.
- Object 6: A holy relic.
- Object 7: A fireproof coat.
- Object 8: A sword.

That'll do for now. Next choose Messages from the main menu and well insert the following as descriptions of the objects.

Message 0: The torch emits a beam of light.
Message 1: In the bag is:
Message 2: It's a dragon slayer's cookbook.
Message 3: It's going a bit soft.
Message 4: It's a 'Rivendeli Cheese and Pickle Sandwich'
Message 5: A delicious slice of cheese.
Message 6: It's a bowl for holding sacred stuff in.
Message 7: It looks like an ACME fire proof coat.
Message 8: The sword is inscribed 'Dragon Slicer'.
Nice.

The object description messages don't have to match the object numbers. I just prefer things that way.



```
I am on the shore of a beautiful  
lake. I can go south to a  
ruined temple or west and back  
to my cottage.  
  
What next?  
I  
I have with me:  
A pocket torch. (lit)  
  
What now?  
DROP TORCH  
I don't have one of those.  
  
What next?  
Z
```

At the moment if you test the game and try to drop the torch (it's carried by the player at the moment), you are told "I don't have one of those." You can't examine it yet either. For that we need to tell PAW what words apply to which objects. So in the next part we'll be looking at how the player can interact with the objects and the locations. In the meantime you could download copies of the PAW technical manuals from Colin Woodcock's site.

<http://www.woodcock88.seriouslyinternet.com/zxf/Resources/ZXPAW/zxpaw.html>

The Code Shed Guide to: BASIC - Part II

Welcome to part 2 of this guide! Hope you've the concepts of Part 1 all safely tucked away in some corner of your brain, 'cos we'll be using some of it (the concepts, not the brain) in this edition. In case you are rusty with some of the concepts, it would be worth your while to quickly go over Part 1 and brush up on things before proceeding further.

Redefining Keys

Okay, with that out of the way, let's head on to some more exciting things in the game we are developing. First up is the key redefinition module. Here is the code first then I'll explain the bits in detail:

```
8400 REM *** Redefine keys ***
8405 LET k$(1)="": LET k$(2)="": LET
k$(3)="": LET k$(4)=""
8406 PAUSE 10: FOR f=1 TO 50: NEXT f
8410 PRINT AT 15,7: INK 7;"Press key for
left": PAUSE 0
8420 LET k$(2)=INKEY$
8430 IF k$(2)="" THEN GO TO 8420
8435 BEEP 0.1,20
8440 PRINT AT 15,7: INK 7;"Press key for
right": PAUSE 0
8450 LET k$(1)=INKEY$
8460 IF k$(1)="" THEN GO TO 8450
8465 BEEP 0.1,20
8470 PRINT AT 15,7: INK 7;"Press key for
up": PAUSE 0
8480 LET k$(3)=INKEY$
8490 IF k$(3)="" THEN GO TO 8480
8495 BEEP 0.1,20
8500 PRINT AT 15,7: INK 7;"Press key for
down": PAUSE 0
8510 LET k$(4)=INKEY$
8520 IF k$(4)="" THEN GO TO 8480
8525 BEEP 0.1,20
8530 PRINT AT 15,5: INK 7;" Keys
Redefined!": PAUSE 100
8540 RETURN
```

If you remember, if the user selected option 2 from the main menu, that would call the key redefinition sub-routine at line 8400 and the above code lists the routine in its entirety.

First in line 8405 we reset all values in k\$ to null and so over-write any previous data in them - starting from a clean slate so to speak.

Line 8406 is an interesting one. It actually does nothing more than delay proceedings. It does this in two ways. First it **PAUSE**s program execution by 1 second and then it further slows things down by going into a loop for another half a second or so (approx). What's a loop you ask? It's just a simple mechanism of repeating whatever you want for a limited number of iterations.

In BASIC, this is done via the **FOR-NEXT** statements. The **FOR-NEXT** (known so since every **FOR** must

have a matching **NEXT**) has the following format:

```
FOR variable = start value TO end value
< BASIC statements that need to be repeated >
...
NEXT variable
```

This is how it works: The *variable* is used to keep a running counter with its initial value being the *start* value. The statements between the **FOR** and the **NEXT** are repeated until the counter hits the *end* value whereupon BASIC wearily moves on the statements following the **NEXT** statement.

For example, let's say you had to print out all the numbers from 1 to 100. Using a **FOR-NEXT** loop you could do this very simply via:

```
FOR f = 1 to 100: PRINT f: NEXT f
```

The logic is quite simple. Since the **FOR-NEXT** combo automatically iterates f through values 1 to 100, we simply print the value of f.

I mentioned that by default the **FOR-NEXT** statement increments the counter by 1. You can change this behaviour so that the counter is incremented by some other value. This is done by tagging a **STEP** qualifier to the **FOR** statement, like so:

```
FOR variable = start TO end STEP increment
```

Note that *start*, *end* and *increment* values can be negative values too. However, keep in mind that if the *increment* is negative then *start* value should be greater than *end* value. For example if you wished to print the numbers from 1 to 100 in descending order, you could do that simply by:

```
FOR f = 100 to 1 STEP -1: PRINT f: NEXT f
```

Well in line 8406, we are doing nothing between the **FOR-NEXT** pair, so BASIC listlessly whiles away its time doing nothing but iterating through the value of f 50 times.

But why are we delaying BASIC? The reason is that we want to give sufficient time for BASIC to recover reading keys from the main menu sub-routine before we start reading keys again in the key redefinition sub-routine. If we didn't do this, things might happen so fast (who said BASIC is slow?!) that as soon as

the user presses option 2 in the main menu, BASIC jumps to the key redefinition sub-routine and reads in the key again (which will be 2 in this case). By slowing things down a bit we are letting Speccy take a quick breather before plunging on.

In line 8410, we print a quick message to tell the user to press the key he desires to move left and then we wait till he does so via the **PAUSE 0** statement.

Once the user presses a key we hit line 8420 where we quickly assign the key pressed (got via **INKEY\$**) to the second element in k\$ - that is k\$(2). Note that our first element is the key to move right - you could change the order of statements so that the right key is read re-defined first.

Line 8430 performs a sanity check to see if k\$ actually holds some value because **INKEY\$** can return a null value if no key is pressed. This is a superfluous statement in the current context because I've used **PAUSE 0** to wait for key presses anyway so **INKEY\$** will return some key value. Anyhow, it doesn't harm to have the check in there but feel free to remove this if you desire.

If found all and good we emit a short **BEEP** to let the user know that the key has been redefined. Similarly, the other keys are likewise re-defined and now the four elements in k\$ will hold the value of the four direction keys that the user has chosen. We'll see shortly how we will use them to move the character around depending on which key the user has pressed.

Creating the graphics

Okay, let's break up the coding monotony by creating graphics for the game. There are 7 separate sets of graphics that we require - one for Krapz the alien, four for explosion effects, one solid block, and one to represent the quanta particles. Sinclair BASIC allows you to create up to 21 UDG's (User Defined Graphics) for your BASIC needs. If you want more in BASIC you'll have to think about redefining the entire character set (which is 96 characters). However, for most purposes 21 is quite enough and in our case 7 is all we need.

BASin has a nifty UDG Editor, which you will find in the Tools menu. We'll use this to create our UDG's (I'll call them sprites from now on). Since there is no way I can explain creating the sprites from scratch via this tutorial, I'll explain

how to modify the existing sprites of the game. The process is the same in either case.

Run the game, and when the menu comes up, press ESC to break into the game - do NOT do a Force Break from BASin (this clears the charset). Bring up the UDG Editor from the Tools menu.

You should see the Editor window listing all the 21 characters, and the first seven should list the sprites being used in the game. On the left of the editor window is the magnified view of the current sprite being edited. Click on the squares to set or reset them. You can edit any of the 21 characters this way. Once you've created/modified the graphics to your hearts content, it's time to get them into the game. What you have to do is change the data in memory by manually POKEing in the new data of your sprites. Don't Panic! BASin makes it easy for you by doing all the hard work, so we'll let it take care of the tedious detail. Here is how:

1. Click on the Send option on the UDG Editor menu and select "As BASIC data"
2. Put 7 in the "number of chars to send" box and 1 in the "chars per line" box.
3. Tick the "Include POKE code" box
4. Put 6060 in the "start at" box and 5 in the "step" box.
5. Review the numbers and press Okay.

BASin will now generate the BASIC statements required to modify the UDG data in the memory automatically! Scroll to line 6060 to see the results. So, how do you print the sprites on the screen? Simple, just switch to Graphics mode and press the alphabet corresponding to the sprite you've created. For example, our Krapz can be invoked by going into graphics mode and pressing 'A'. If you are wondering how to get into Graphics mode in BASin, you can set the key for this in BASin by going to Option>BASin>Programming Aids>Graphics Modifier and selecting the key you want. Mine is set to go into Graphics mode whenever I press the Scroll Lock key.

That's all there is to it, really, as far as designing the sprites for the game goes.

In the next edition of Code Shed, we'll take a look at how to animate the sprites we have designed and make things move around and explode! Stay tuned!

In this column, we look at classic games from years gone by and see how they hold up against later competition. For this first feature, we crank the year back to the ZX Spectrum's debut year of 1982 to look at the games that were released then and see how time has treated them.

Football Manager

(Addictive Games)



It's a goal! Listen to the crowd bleep

Being written entirely in BASIC, this was a game that never really impressed anyone from a technical standpoint. Still, machine code snobbery was yet to arrive on the Spectrum scene, and word of mouth enthusiasm about how the game played was enough to boost sales to a staggering 300,000 copies. So, how does it hold up now? Pretty well, I'd say.

It's fairly bereft of options but author Kevin Toms achieved that delicate balance that gives you a real feel of involvement in your team's fortunes yet doesn't overwhelm you with options that are critical to your success: You pick and choose your team from your squad, arrange transfers, possibly take out a loan to buy more players and that's that.

The highlights, although widely ribbed at the time for their simplicity, still deliver the tension as you have to wait and see whether your players can by turns pull off a miraculous victory or throw away a huge advantage. It still leaves you wanting more, after every match and I think that's all very faithful to the sport of football.

Verdict: A much copied formula that's been very much embellished in later games, but this still has the basics.

Flight Simulation

(Psion)



Fly off into the wild blue yonder

Another top seller that dominated the charts for nearly a year, this game brought the feel of flying a light aircraft to the home computer. It scored well against the early competition too, having reasonably fast responses, and a good graphical view of the horizon with a few scenic lakes to fly around as well as the obligatory runways.

Today, it looks very basic when you compare it against the top PC flight simulators though, and even some of the ones that appeared in the latter years of the Spectrum with only a small control panel and a few token instruments. The lack of much in the way of flying options also serves to limit the game play too. Once you've mastered the arts of taking off and landing, that's pretty much it.

During the next few years, a whole host of combat flight simulators came onto the Spectrum scene and ousted this game from the front line although it was always regarded, alongside Digital Integration's Fighter Pilot, as one of the more faithful flying simulations.

Verdict: Still a reasonable simulation, if very simplistic, compared to subsequent releases. Gets a trifle dull without combat or serious navigational challenges.

Penetrator

(Melbourne House)



Deep into enemy territory the colour clash just gets worse

Philip Mitchell must have been one of the busiest programmers in the early years of the Spectrum. Between creating *The Hobbit* and *Sherlock*, he turned his hand to this game which was one of the first games of true arcade quality to come out on the Spectrum. Based closely on the arcade hit *Scramble* it added radars that slowly track you down until you're bombarded with a barrage of accurate missile fire. Best of all, the game also came with an editor that allowed you to change the landscape and the distribution of enemy missiles and radars allowing for pretty much unlimited game play.

The graphics have stood up rather well with the line drawn landscape and nicely animated radars still looking good. There is a lot of colour clash though, as radars tend to leave their attributes behind them colouring in the landscape as it scrolls across.

The game also seems quite short these days, and rather easy although that can be remedied somewhat by using the editor. My last grip is the odd choice of keys (P doubles up as both right and fire) which don't exactly sit easily, although given that there weren't any joysticks with two fire buttons in those days that can almost be forgiven.

Verdict: An elegant scrolling shooter that still plays just as well as ever.

Time Gate

(Quicksilver)



In Space, nobody can hear the sound effects

I remember this being pretty much the star of the Spectrum scene when it first came out at the end of 1982. *Penetrator* was good, but that was just a 2D game and a fairly obvious copy of an arcade favourite. Instead, *Time Gate* offered a 3D moving star field, with an interrupt driven engine roar continually blasting away in the background, whilst you fought furious dogfights with alien spacecraft. All this against a back story of time travel in a desperate mission to save the human race by wiping out the alien planet before they'd even reached space!

Looking at it now though, the game seems terribly dated and simplistic. Graphically, the 3D is only a fudged affair: The star field moves in fixed lines, the alien spacecraft move around in character-square sized jumps, and the perspective of your ships weapons just seems wrong. The rope-mat planets just look downright awful, and your spacecraft cockpit looks uncannily like an 80's clock radio. There's no real depth to the game play either. You just warp through the sectors until you find the time gate; and sometimes you don't even have to fight a single alien spacecraft to complete the game.

I'd like to find something positive to say about this game, as it was a real favourite all those years ago, but that's hard work. About all I can think of is the graphical hyperspace effect as the stars stretch out into lines, reminiscent of the *Millennium Falcon's* jump in *Star Wars*.

Verdict: Time has done this game no favours.

Arcadia

(Imagine)



Are those the arrows of outrageous fortune?

Imagine were one of the first big names on the Spectrum games scene and this was their very first hit. Another game aspiring to arcade quality, this fast and furious multi-level shooter cleared the decks of tired space-invader clones and delivered the message that the Spectrum was a games machine to be reckoned with. It gave the spectrum's sound capabilities a thorough working over and surviving the twelve levels were a serious challenge.

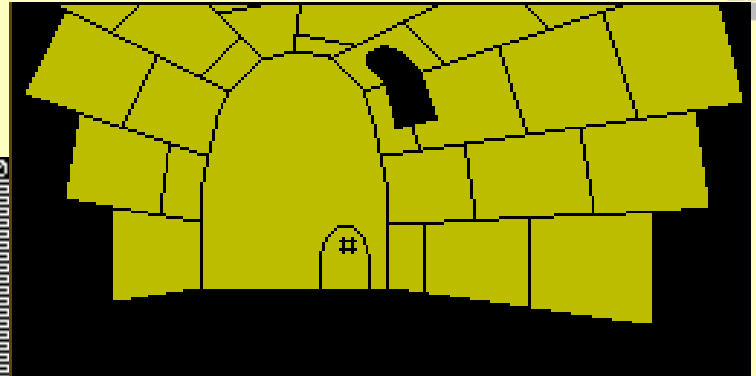
Nowadays, the graphics look very poor with vast amounts of flicker and colour clash. Only in the environment in which the game was launched, where UDG-figures were the standard to which most game graphics went, did this ever look good. That said, the game still plays fairly well, although on a lot of the levels the only way to play is to shoot out all but a couple of the aliens and hide in the corner whilst the time runs down.

Verdict: A reasonable shoot-em-up for its day, but very much eclipsed by later games.

The Hobbit

(Melbourne House)

This was the first major commercial tie-in for the Spectrum and unlike many later efforts was universally lauded for both its faithful translation of the book to the computer game format, but also for its innovative parser that went beyond the standard verb-noun text adventure convention for the time and the interactive characters such as Thorin and Gandalf that populated the game.



```
The nasty goblin captures you.  
You are in the goblins dungeon  
=====  
YOU CANNOT OPEN THE SMALL  
INSIGNIFICANT CRACK.
```

```
> WAIT  
> D  
+
```

Oh no! Not the goblins dungeon again

Looking at it now, it's a fairly small adventure compared to later games on the Spectrum scene, especially Level 9's giant blockbusters with their hundreds of locations and rambling text descriptions.

By contrast, the Hobbit relies a lot for atmosphere on the book with only a small amount of text for each location. The graphic illustrations aren't brilliant for the most part and are also rather slow to draw which leaves you waiting a lot, especially when you pass through the same places often; the worst of which is illustrated above. The parser, although reasonably advanced, is also very finicky with you having to type in some long and complicated sentences to do fairly simple things sometimes.

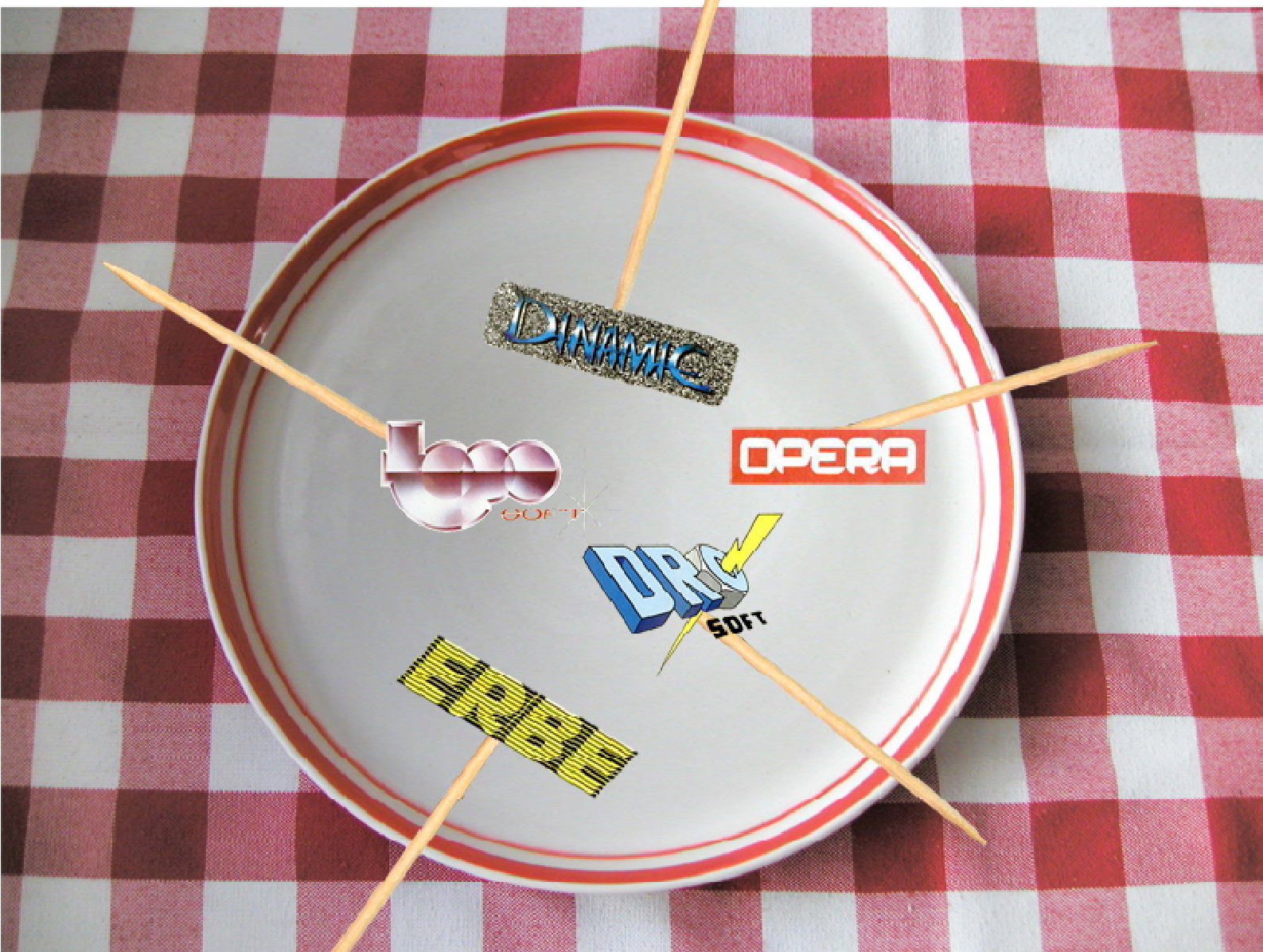
There's also an irritating randomness to proceedings: Thorin and Gandalf will sometimes just decide to be uncooperative and escaping from the dreaded goblins dungeon without getting immediately recaptured requires a lot of luck.

It also has to be said that it's been over a decade since text adventures were a commercially viable genre.

With that out of the way though, I'm still fond of this game as it introduced me to text adventures and would still serve as a good game for beginners, especially those who enjoy the works of Tolkien, as you can get a long way in it fairly easily and it does tie in nicely with the book.

Verdict: Not the greatest text adventure by a long way, but still a good introductory game for the genre.

SPA2IN IS DIFFERENT



<http://spa2.speccy.org>

A STOP
NEW

Do you own any Spanish tape or disk?
Did you know they're endangered species?
Help us to preserve them!



New design, more information.
Further possibilities. Same goal as ever.
Discover the new SPA2.
Soon in your favourite browser!

SPA2

SPANISH SPECTRUM ARCHIVE

A site hosted by

SPECCY.ORG

Richard Jordan of Retrospec has kindly agreed to answer a few questions regarding the recently released Atic Atac

ZXS: Richard, thank you and hello. Have you had time to download and read a copy of ZX Shed yet?

R: I have to admit I'd never heard of ZX Shed until your email arrived, but I've just had a shuftly at the first two issues and wow, what can I say? It certainly makes coding remakes look like a walk in the park, you guys must be putting some serious effort in. Good on yer.

ZXS: Thanks very much. I've always been impressed with the games available on the Retrospec website. Part of the appeal is seeing an old favourite brought right up to date. How do you become interested in doing remakes?

R: I originally became interested in remakes after playing Andy Noble's Manic Miner. For me, it's still the perfect remake and I'd like to think I've come pretty darn close to hitting the same balance of originality and authenticity.

ZXS: Yeah that's one of my favourites too. I'm often torn between playing Andy Noble's PC version or loading it up in SPIN and playing the original for true nostalgia. Was writing the code for the Atic Atac remake especially difficult and how long did it take?

R: It was a simple game on the Speccy and it was an easy game to convert to the PC. I've done remakes before and Atic Atac has proved to be the least challenging. That's not taking anything away from the game though; the beauty of it is in the game play.

Time wise, It's hard to put a figure on it. I started in December 2004, but it didn't get released until May 2006. That's not to say it took 18 months of effort because it didn't, not by a long shot. A lot of the delay was down to my life and Mr Blythe's life getting in the way. If everyone involved had worked on it 8 hours a day non stop then I reckon we could have got it all done in something like 8 weeks. A bit of a guess really but I'm just trying to give you an idea of how lazy I get!!

ZXS: What did you use to code the remake? Was any specialist software used by the team to produce the graphics, music and sound effects?

R: It's coded using VC++ 6.0 and uses the excellent Allegro games library and the equally competent FMod sound library. I don't actually know what the other guys use to do their bits. Both JB on the pixels and Matt on the music are professionals at what they do so I guess they use industry standard stuff.

ZXS: One of the things I like best about Atic Atac is that it's been brought up to date but still manages to look, feel and play just like the original.

R: I paid a lot of attention to detail, making sure

that the game's essence survived the translation. Judging from the feedback so far, it's not been a bad effort!

ZXS: You said that it wasn't an especially hard game to convert. Was there anything that caused any problems? Do you recall anything that was particularly challenging?

R: My biggest worry was the sound effects. Initially it was going to be all new, but that just didn't work. The decision to keep all of the original sounds (with a touch of reverb) and enhance the overall effect with brand new bits and bobs worked a treat.

ZXS: The sounds work well and I really fell in love with the music during the introduction. I also like the way that everything has enhanced but can instantly be recognised as part of the original game.

R: Matt Simmonds came up trumps on the music front, although I guess it wasn't that hard for him because he's a living, breathing music factory. As for the look of the game, John Blythe's graphics are better than ever and he had an awful lot of work to do. Poor thing, I had to nag him a bit during the final weeks of the project.

ZXS: Ouch, you sound a bit like our editor. How do you feel about the finished article?

R: Overall, in my opinion, I think the game works because JB's graphics look great, the music and sound effects are just right and the game play is virtually identical.

ZXS: Did you ever do much programming for the Speccy?

R: Yep, sure did, I wasted many hours writing pretty crap games in Speccy Basic. Towards the end of the 8-bit era I started dabbling with machine code on the 128k +2, my proudest moment being an interrupt driven 3 channel music player. Magical stuff.....well, I thought so anyway!!

ZXS: Sounds good to me, although never progressed much further than basic. Richard, thanks for talking to us and I hope you get time to read this in and the review in issue three. Before you go, what will you be working on next and which Speccy game would you most like to do a remake of?

R: There are no firm plans for another game at the moment, but I'd like to do something isometric. I have an 80% finished remake of Fairlight sitting on my hard disk gathering dust, but The Edge got a bit funny about the project so it got canned. That aside, other games I'd love to rewrite are The Great Escape, Ranarama and Underwulde.

Welcome to yet another instalment of Lost Treasures, the article that looks at games that slipped under the Speccy games radar. The two games I'm looking at this time both include archaeologists as the heroes. I thought it was an interesting choice as I'm currently studying this in university.

The Archaeologist



The first of these is The Archaeologist (oddly enough) by Sneesby Software. This game was included on Spectrum Computing #18, a tape-based magazine published by Argus Press Software in 1985. You play the part of an archaeologist travelling through underground caverns in search of treasures and artefacts.

It is essentially a nice, fast-moving clone of Jet Set Willy where you have to jump from platform to platform, avoiding nasties (some of which move much faster than you) and collecting items which are dotted around the screens. Overall, this is a fun little arcade platformer that should keep you occupied for a good while. Just don't be fooled into thinking that real archaeology is as exciting as this.

Steven Spielberg has a lot to answer for. Unfortunately, this game contains a few bugs. The biggest one I noticed was using the Super-Jump key (which was mentioned on the title screen) either does nothing or kills you when you press it. It's not really a problem though as I found I didn't need to use it while playing.

Babaliba

Spanish software house Dinamic has a reputation of producing very difficult games. Babaliba is one of these, although not quite as difficult as some of their other games (Army Moves, Freddy Hardest etc.), it is still quite a challenge. Originally published in Spain in 1985, it was re-released for the rest of us afterwards by Silversoft. It is the 2nd part of a trilogy that includes Saimazoom and Abu Simbel Profanation.



You play the part of heroic archaeologist and explorer Indiana Smith (I wonder if Spielberg knew about that little copyright infringement and if there were any legal problems for the company as a result) who has to rescue his friend Victor plus try and find the treasure which is hidden in a palace somewhere in the Gardens of Babaliba. There are three keys hidden in the maze which he has to find to gain access to the palace. There are also lots of creatures roaming the maze but our hero has bombs which he can drop to take care of the patrolling nasties. While playing the game, I found I didn't have to use the bombs as if you see a creature blocking your path, you can simply leave the screen and re-enter and it will move to a different part of the screen. Like I said earlier, this one is a bit easier than other titles from Dinamic. I'm definitely going to play this a bit more to see if I can finish it.

VULCAN - THE TUNISIAN CAMPAIGN

Vulcan is a strategy game, originally coded by R.T. Smith. As a battlefield strategy simulation, R.T. Smith set the standard for strategy games for years to come. It is easy to see why. It is nearly limitlessly playable in its original format alone, offering the player the chance to order five historical armies into combat in any way he sees fit. If both sides follow historical methods of attack, the outcome is historical - a very reassuring feature!



The game recreates the fighting that happened in Tunisia in 1942/43 between two of the world's most famous Field Marshalls - Rommel with his Africa Corps, and Montgomery with his 8th Army. Italian, French, and American armies are also included. Kassering pass, Eilbote, and Operation Vulcan can all be relived.

The player is put in the place of a commander, sitting at his table with all those pieces representing troop formations that we've all seen them push around with long sticks on all those old war films. The screen shows a scrolling map of Tunisia, laden with units which appear as little rectangular 'pieces.' Their colours and graphics represent their unit equipment and nationality.

Each turn is one day of the campaign. During your turn, each unit flashes in sequence so you can give them their day's combat orders. You can move a unit, move in assault mode, pack up and travel fast, stay put, or stay put and fortify its position. After all the units have been ordered, the computer thinks or your human opponent then has his turn.

Only after all the day's combat orders have been issued does the computer go into 'Forces Moving' and those orders are carried out in real-time. If enemy units meet, combat takes place. Units flash red as they take damage, and the sound of machinegun fire rattles across the battlefield. The outcome of combat is dependent on many variables, such as how many tanks or men your units have the terrain they are on,

the weather etc. In fact everything that affected combat in reality is taken into consideration. One thing is for sure, and that's not to let your unfortified infantry get caught by tanks in open terrain!

So, it makes me pleased to announce that you can now recreate any battlefield situation using 'Vulcan' and the editing guide that I've created. Ever wanted to know what would've happened had the French remained Vichy? Or maybe the Africa Corps had not been beaten at El Alamein? Maybe you just wanted to go mad and give the French forces 50 armoured divisions... now it is all possible. Tunisia will never have seen such bloodshed. It's worth it to give it a shot, so enjoy my guide!

Welcome to the battlefield simulation tutorial. This tutorial describes how to make limitless different simulations of battles on a Tunisian map, using the Spectrum War-game 'Vulcan,' by RT Smith. 'Vulcan' is a classic war-game which to this day more accurately represents what would happen if you were commanding your troops in a WW2 style war room.

For this tutorial I am assuming you have already played 'Vulcan' before, and are familiar with all the rules of the game. Instructions and the game itself can be found at www.worldofspectrum.org. Knowledge of 'poke' software that is usually bundled with Spectrum emulators is also assumed.

To edit any unit is simple. Every unit in the game has its own section of memory addresses that control everything about it.

The most basic and useful modification you can make, is to change the number of men or tanks a unit has. To make it easier, a complete list of memory addresses for unit strengths is included here. Any unit can have strength between 1 and 255. Even a battalion could have 255 tanks! Remember, infantry are counted in multiples of 30, so setting an infantry unit's strength to '100' gives it 3000 men.

For example, if you didn't like the default number of tanks the 7th panzer regiment has (96,) and wanted to change it mid game to 120, you would 'poke' memory address 53221 to 120 bytes. This does not update the debriefing counter.



The value just after the one for this (in game) strength is the one for its initial starting strength on the map. It does not work once the unit has appeared. This does update the debriefing counter, (whereas poking mid game does not.)

You can have endless possibilities by editing the strengths of existing units alone, but making other modifications to completely change a unit into something else is a little more complex.

Changing a unit's equipment type, nationality, and its 'stacking' size are all done at the same memory address. The address to do this for any unit is always located the one before the one for its strength. In our example of the 7th panzer regiment, it would be at address 53220.

There are 102 different unit possibilities that we could change the 7th panzer regiment into. There are 6 different nationalities, (one is a hidden army which didn't actually appear in the game!) each with 17 different unit types. A list of all the different possibilities is included with this tutorial.



For example, if I wanted to change the 7th panzer regiment into a French parachute division I would poke 53220 to 157.

Please be aware that unit type controls many aspects of a unit, and even if you modify these aspects, they will default back in the next turn of the game. For example a medium tank unit is intended to have 10 moves per turn, so modifying moves per turn lasts one turn only.

To place a unit anywhere on the map is controlled by the addresses two and three places after its strength. They are grid coordinates. The first address controls spaces on the map from the left, the second controls spaces on the map from the top.

So, to automatically teleport your new French Para infantry division to the position you want it in for your new scenario you'll undoubtedly be making, you need to poke addresses 53223 and 53224.



4th from strength is how many moves per turn a unit has. Values range between 0 and 63. Poking 53225 to 63 will let your new unit travel all the way across the map in one go! This lasts one turn only. Poking to 64 resets the counter back to 0, (so you have 4 possibilities up to the maximum 255 bytes to set the same MPS.)

5th from strength controls the position a unit is in to receive supplies. This lasts one turn only. 6th from strength controls how many days a unit has been fortified for.

Values range from 0 to 255. If you set the value to 255, then fortify the unit some more while playing the game, the new fortifications will have no further effect. This can be set ready before a unit appears.

7th from strength controls the unit's position within its divisional stack.

It has no real use - the game calculates the appropriate values automatically 8th and 9th from

SPIN FEATURES

strength control the grid coordinates of the flashing cursor when controlling a unit in game. Modifying this has no use.

10th from strength controls a unit's supply level. Values range from 0 to 255. In the game, 50 is 'very good' supplies, and is the maximum amount of supply the simulation will ever give. However, you can poke the supply address greater than 50, giving supply overload. Poking 53231 to 255 would give your new unit a near limitless excess of supply!!!



One of the most important modifications you can make is located 11th address from the one for strength. This is the one that controls the quality of the unit's fighting ability. This address controls both morale and attack modifier of a unit. Values range from 0 to 255, and are modified in blocks of 16. Each block of 16 is for one morale setting. For example, values between 0 and 15 give morale of 0. Values between 16 and 31 give morale of 10, 32-47 is morale 20, etc. Values in between the blocks of 16 control the unit's attack modifier for a corresponding morale. For example 0 gives AM of 0, 1 gives AM 10, 2 gives AM 20, etc. You get the picture! The quick reference list included here gives the most common values.

+13 from strength controls which division a unit belongs to. A quick reference list is included here.
+14 from strength modifies the command highlighted in the unit menu. This is a more important value than it seems. When you create a new unit this defaults to 0, a corruption value, so it must always be set manually to the 'Hold' orders value of 3 at the same time as the unit is created.

+17 controls the number of turns into the game before a unit will first appear.

Last but not least is a unit's efficiency. The value is located 18 addresses from strength. Values can range between 0 and 255, but setting efficiency

greater than 100% lasts one turn, then corrupts the unit. The maximum efficiency the game would usually allow is 100%, so 127% puts your men on steroids. Once that extra efficiency is lost in combat, it cannot be regained (unless you go pokey again!!!) Values from 1 to 100 are for efficiency's considered without combat experience, 128 to 228 are with combat experience.

You may wonder what happened to the descriptions of all the addresses that I left out before we got to +18. Well, stop wondering, or you'll corrupt your game!!!

Unfortunately, due to memory overflow, it is not possible to add a greater number of units than the game already allows, so you can only modify existing units into new ones. If you are programming your new scenario early in the game, you can 'pull' units into the game early, or you can retrieve destroyed ones later on.

There are a number of methods to bringing new units into the game. The safest and easiest is to choose a unit which hasn't appeared yet. Set the unit's counter for how many turns until it appears. Set it to the turn after the one you are playing, then on the next turn it will appear ready for editing with all its correct values intact! You can then edit it into whatever you want!

The third value from strength, the one that controls the horizontal grid coordinate, is of course irrelevant before a unit has made its appearance. So, before then, it is used to control where a unit will appear when it does. There are many more spawn points than the ones used in the official game. Please refer to the table included here.

To bring a destroyed or withdrawn unit into the game, you will need to set ALL the values for that unit manually, as most will have defaulted to 0. If you forget to set a value, the unit will automatically be destroyed, or unmovable, or the game will corrupt, or, the game defaults a unit that isn't supposed to be there back out of the game on the next turn. To keep the unit, you need to re-set its grid co-ordinates on the next turn, and (for some reason) the game doesn't try removing it a second time. Its there to stay... Or If you have brought a unit in early using the latter method, then it will automatically be destroyed on the date it is actually due to appear, calculated as men lost, and points deducted.

Basically, set ALL values manually unless you want to use the simple method of altering a unit's start date before messing around.

Another warning, if you try to set all values manually to bring back a withdrawn unit, it is impossible if the unit was supposed to have been withdrawn by the

game, but you edited it out of the game before its withdrawal date thus preventing the withdrawal. To prevent a unit appearing in the game in the first place is also easy. Simply set its starting strength to 0.

Many new scenarios using the editing methods described below, and more, are available by e-mailing me at danielthorlby@hotmail.com and will be available for download shortly. Also, a programmer from London called Jason Kelly is currently working on an official remake of the game for the windows XP platform, release date to be announced.

UNIT (IN GAME) STRENGTH ADDRESSES

ALLIES

50971 - HQ 78th infantry Division
50996 - 36th Infantry Bde
51021 - 11th Infantry Brigade
51046 - 1st Guards Brigade
51071 - HQ 6th Armoured Division
51096 - 38th Irish Bde
51121 - 26th armoured brigade
51146 - HQ 46th Midland Division
51171 - 139th infantry Brigade
51196 - 128th infantry brigade
51221 - 130th infantry Brigade
51246 - HQ 1st Infantry Division
51271 - 24th Guards Brigade
51296 - 2nd Infantry Brigade
51321 - 3rd Infantry Brigade
51346 - HQ 4th Mixed Division
51371 - 12th Infantry Brigade
51396 - 10th Infantry Brigade
51421 - 21st Army Tank Brigade
51446 - V Corps HQ (British)
51471 - IX Corps HQ (British)
51496 - 25th army tank brigade
51521 - 1st Parachute Brigade
51546 - 6th Commando Battalion
51571 - Hart Force
51596 - 1st Parachute Battalion
51621 - 3rd Parachute Battalion
51646 - Blade Force

8th ARMY

51671 - HQ 51st Highland Division
51696 - 152nd Infantry Brigade
51721 - 153rd Infantry Brigade
51746 - 154th Infantry Brigade
51771 - HQ 7th Armoured Division
51796 - 22nd armoured brigade
51821 - 4th Light armoured brigade
51846 - 131st Lorried Infantry Brigade
51871 - HQ 2nd New Zealand Division
51896 - 5th NZ Infantry Brigade
51921 - 6th NZ Infantry Brigade
51946 - HQ 1st Armoured Division
51971 - 1st Armoured Brigade
51996 - 7th Motor Brigade

52021 - HQ 50th Northumberland Division
52046 - 69th Infantry Brigade
52071 - 151st Infantry Brigade
52096 - HQ 4th Indian Division
52121 - 5th Indian Brigade
52146 - 7th Indian Brigade
52171 - HQ 56th London Division
52196 - 167th Infantry Brigade
52221 - 169th Infantry Brigade
52246 - 201st Guards Brigade
52271 - 8th Armoured Brigade
52296 - 23rd Armoured Brigade
52321 - L Force
52346 - XXX Corps HQ (British)
52371 - X Corps HQ (British)

AMERICAN ARMY

52396 - HQ US 1st Armoured Division
52421 - CCA
52446 - CCB
52471 - CCC
52496 - HQ US 1st Infantry Division
52521 - 16th Infantry Regiment
52546 - 18th Infantry Regiment
52571 - 26th Infantry Regiment
52596 - HQ US 9th Infantry Division
52621 - 39th Infantry Regiment
52646 - 47th Infantry Regiment
52671 - 60th Infantry Regiment
52696 - HQ US 34th Infantry Division
52721 - 168th Infantry Regiment
52746 - 133rd Infantry Regiment
52771 - 135th Infantry Regiment
52796 - II Corps HQ (US)
52821 - 503rd Parachute Bn

FRENCH ARMY

52846 - XIX Corps HQ (French)
52871 - Division Du Constantine
52896 - division du-Maroc
52921 - Division D'Alger
52946 - Division D'Oran

AXIS

52971 - XC Corps HQ
52996 - HQ Division von Broich
53021 - Witzig Engineers
53046 - 5th Para Regt
53071 - Barenthin Regt
53096 - Marsch1
53121 - Marsch2
53146 - Marsch3
53171 - Marsch4
53196 - HQ 10th Panzer Division
53221 - poke for 7th panzer Regiment
53246 - 86th Panzergrenadier Regt
53271 - 69th Panzergrenadier Regt
53296 - Kampfgruppe Witzig
53321 - Superga
53346 - 50th Special Brigade
53371 - 10th Bersaglieri Regiment

AFRICAN FRONT

53396 - 47th infantry regiment
53421 - 190th Panzer Bn
53446 - 501st Panzer Bn
53471 - 160th infantry regiment
53496 - HQ 334th Infantry Division
53521 - 754th infantry Regt
53546 - 755th infantry Regiment
53571 - 756th Mountain Regiment
53596 - HQ Hermann Goering Division
53621 - Hermann Goering Para Regt
53646 - Hermann Goering Panzer Bn
53671 - 165th Infantry Regt
53696 - 999th Penal Division

AFRIKA KORPS

53721 - Deutsches Afrika Korps HQ
53746 - Corpo d'Armata XX
53771 - Corpo d'Armata XXI
53796 - HQ 15th Panzer Division
53821 - poke for 8th panzer Regiment
53846 - 115th Panzergrenadier Regt
53871 - HQ 21st Panzer Division
53896 - poke for 5th panzer Regiment
53921 - 104th Panzergrenadier Regt
53946 - HQ Centauro Armoured Division
53971 - 131st Armoured Regiment
53996 - Trieste Division
54021 - La Spezia
54046 - Pistoia
54071 - GGFF YOUNG FACIST
54096 - SV288
54121 - Sahara Group
54146 - 164th light Division
54171 - 90th Light Division
54196 - Ramke

UNIT TYPE CODES

BRITISH

9 - British medium tank battalion
10 - British churchill tank battalion
11 - British motorised infantry battalion
12 - British foot infantry battalion
13 - British Parachute Infantry Battalion
14 - British foot HQ
15 - British motorised HQ
73 - British medium tanks brigade
74 - British Churchill tank brigade
75 - British Motorised Infantry brigade
76 - British foot infantry brigade
77 - British parachute infantry brigade
137 - British medium tank division
138 - British churchill tank division
139 - British motorised infantry division
140 - British foot infantry division
141 - British parachute infantry division

AMERICAN

17 - American Medium tank battalion
18 - American Churchill tank battalion
19 - American Motorised Infantry Battalion

20 - American foot infantry Battalion
21 - American Parachute Infantry Battalion
23 - American Motorised HQ
81 - American Medium Tanks Brigade
82 - American Churchill tanks brigade
83 - American Motorised Infantry brigade
84 - American foot infantry brigade
85 - American parachute infantry brigade
145 - American medium tank division
146 - American churchill tank division
147 - American motorised infantry division
148 - American foot infantry division
149 - American parachute infantry division

FRENCH

25 - French Medium tank Battalion
26 - French churchill tank battalion
27 - French motorised infantry battalion
28 - French foot infantry battalion
29 - French parachute infantry battalion
30 - French foot HQ
31 - French motorised HQ
89 - French medium tank brigade
90 - French Churchill tank brigade
91 - French motorised infantry brigade
92 - French foot infantry brigade
93 - French parachute infantry brigade
153 - French Medium tank division
154 - French Churchill tank division
155 - French motorised infantry division
156 - French foot infantry Division
157 - French parachute infantry Division

GERMAN

33 - German medium tank Battalion
34 - German Churchill tank Battalion
35 - German motorised Infantry Battalion
36 - German foot infantry Battalion
37 - German parachute infantry Battalion
38 - German HQ
39 - German Motorised HQ
97 - German medium tanks Brigade
98 - German Churchill tank brigade
99 - German Motorised Inf Brigade
100 - German foot infantry Brigade
101 - German parachute infantry Brigade
161 - German medium tank division
162 - German churchill tank Division
163 - German motorised infantry Division
164 - German foot infantry Division
165 - German parachute infantry Division

ITALIAN

41 - Italian medium tank battalion
42 - Italian Churchill tank battalion
43 - Italian motorised infantry battalion
44 - Italian foot infantry battalion
45 - Italian parachute infantry battalion
46 - Italian foot HQ

47 - Italian motorised HQ
 105 - Italian medium tank brigade
 106 - Italian Churchill tank brigade
 107 - Italian motorised infantry brigade
 108 - Italian foot infantry Brigade
 109 - Italian parachute infantry brigade
 169 - Italian medium tank division
 170 - Italian Churchill tank division
 171 - Italian Motorised Inf Division
 172 - Italian Foot Inf Division
 173 - Italian parachute infantry division

DARK BLUES (axis)

49 - Dark blue medium tank battalion
 50 - Dark blue Churchill's battalion
 51 - Dark blue motorised infantry battalion
 52 - Dark blue foot infantry battalion
 53 - Dark blue parachute infantry battalion
 54 - Dark blue foot HQ
 55 - Dark Blue motorised HQ
 113 - Dark Blue medium tank brigade
 114 - Dark Blue Churchill tank brigade
 115 - Dark Blue motorised infantry brigade
 116 - Dark Blue foot infantry brigade
 117 - Dark Blue parachute infantry brigade
 177 - Dark Blue medium tank division
 178 - Dark Blue Churchill tank division
 179 - Dark Blue motorised infantry division
 180 - Dark Blue foot infantry division
 181 - dark blue parachute infantry division

DIVISION'S NUMBER LIST

1 78th infantry
 2 6th armoured
 3 46th midland
 4 1st infantry
 5 4th mixed
 6 51st highland
 7 7th armoured
 8 2nd new Zealand
 9 50th Northumberland
 10 1st Armoured
 11 4th Indian
 12 56th London
 13 US 1st armoured
 14 US 1st Infantry
 15 US 9th Infantry
 16 US 34th Infantry
 17 Division Von Broich/Mantueffel
 18 10th Panzer Division
 19 334th Infantry Division
 20 Hermann Goering Division
 21 15th Panzer Division
 22 21st Panzer Division
 23 Centauro Armoured Division

QUICK QUALITY REFERENCE TABLE

MOR20 AM30 50
 Mor20 AM40 66
 MOR30 AM50 83
 MOR30 AM60 99

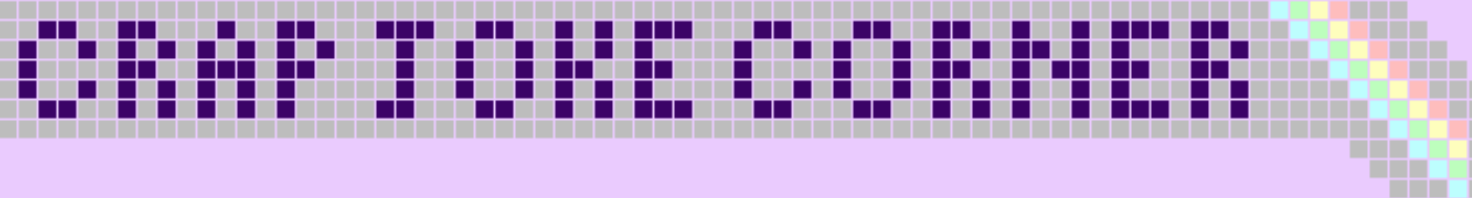
MOR30 AM70 115
 MOR40 AM70 116
 MOR40 AM80 132
 MOR100 AM80 138
 MOR40 AM100 164
 MOR50 AM100 165
 MOR50 AM120 197
 MOR60 AM120 198
 MOR40 AM140 228
 MOR50 AM150 245

SPAWN POINTS TABLE

0 - top left airfield - for anyone
 1 - Tebessa airfield - for anyone
 2 - Tripoli (airfield)
 3 - Tunis - defaults to Bizerta
 4 - Bizerta - defaults to Tunis
 5 - 2nd airfield down on left edge - defaults to top left airfield - must be owned for axis and no default
 19 - Mareth airfield - no default - axis must own
 20 - Sicily
 21 - south of Lekef
 22 - Thala (one square left)
 23 - one square north of Gafsa
 24 - Sicily (again) - defaults to top left airfield
 25 - medjez - defaults to Tebessa airfield
 26 - Tabarka - defaults to fondouk airfield - can only spawn if owned by your army - nodefault
 27 - Thala - default to sicily
 28 - Sbibia - defaults to medjez
 29 - pichon - defaults to tebessa
 30 - Tebessa - no default - must be owned for axis or default to bizerta
 31 - Kasserine - defaults to medjez
 32 - Gabes - defaults to sicily - axis must own
 33 - Medenine - defaults to Fonduok airfield - axis must own
 34 - Very top left of screen
 35 - top left airfield - defaults to top left of screen for allies, bizerta
 airfield for axis (must be owned not to default)
 36 - top left for allies - defaults to Bizerta airfield for axis
 94 tebessa - who can control - must be owned for axis or no default
 95 Kasserine
 96 Gabes
 97 Medenine
 98 Kasserine
 99 medjez

EDITING RULES QUICK REFERENCE

+1 - after (in game) strength - starting strength - can be modified before unit appears
 +2 - after (in game) strength - squares from left - before unit appears controls start position



We ask for articles, and get next to nothing. We ask for your photos & get next to nothing. We ask for letters & get none. We asked for your crap jokes and suddenly we have tapped into your mentality.

Dead Funny

An old lady was very upset as her husband Albert had just passed away.

She went to the undertakers to have one last look at her dearly departed husband. The instant she saw him she started crying.

One of the undertakers strode up to provide comfort in this sombre moment.

Through her tears she explained that she was upset because her dearest Albert was wearing a black suit, and it was his dying wish to be buried in a blue suit. The undertaker apologized and explained that traditionally, they always put the bodies in a black, but he'd see what he could arrange.

The next day she returned to the undertakers to have one last moment with Albert before his funeral the following day. When the undertaker pulled back the curtain, she managed to smile through her tears as Albert was resplendent in a smart blue suit.

She said to the undertaker "Wonderful, wonderful, but where did you get that beautiful blue suit?" "Well, yesterday afternoon after you left, a man about your husband's size was brought in & he was wearing a blue suit. His wife explained that she was very upset as he had always wanted to be buried in a black suit," the undertaker replied.

The woman smiled at the undertaker. He continued, "After that, it was simply a matter of swapping the heads"
Lee Fogarty.

Commodore Bashing

Q. What do you call a Commodore 64 at the bottom of a pit?

A. A good start.
Anon.

More Commodore Bashing

It is the day of the annual World of Spectrum V Lemon 64 football match, and Martijn, the WOS team coach, enters the changing room to find his team are all moping around looking thoroughly unmotivated.

"What's up you guys", he asks, "We're going to give the Commodore guys a good kicking yet none of you seems interested". "That's just it", reply the

team, "None of these Commodore forumites can kick a ball and we always win easily. Where's the fun in that? We'd all rather just be down the pub talking about Spectrum games."

Martijn thinks for a second, then agrees. "Okay, I'll leave MeltheBell to face the Lemon 64 team single handed, while the rest of you go down the pub. Is that okay?" So they all agree and with that the rest of the team head off to the nearest public house for a few drinks and a game of Chaos while Mel prepares to face the Lemon team alone. Half-way through the first game Lee Fogarty rings Martijn on his mobile to ask how the match is going.

"We've been playing for ten minutes and it's going well so far," says Martijn, "Mel's just dribbled the ball from the half-way line past the entire Lemon 64 team and tapped the ball through the goalkeeper's legs to score. We're winning 1-0!" With that the rest of the WOS team breathe a sigh of relief and carry on with their game of Chaos.

An hour and a half later the Spec-chums are on their way back from the pub and Lee calls Martijn again to ask how the match finished. "Okay", he replied, "Lemon 64 grabbed a last-minute equaliser, but we still drew the game 1-1." The Spec-chums agree this is not a bad result under the circumstances, and head off to the changing room to congratulate MeltheBell. When they get there, they find MeltheBell is sitting with his head in his hands, looking distraught.

"I've let you all down guys, I'm sorry," says Mel. "Hey don't worry Mel," they reply, "It was 11 against 1 and you very nearly won. You didn't let us down at all." "No you don't understand," retorts Mel, "I got sent off after 12 minutes..."
Anon

Yellow Belly

I went to the doctors yesterday because I thought I was watching too much television. He asked, "What are the symptoms?" I said they were a yellow cartoon family.

Bethan

Did We Do Commodore Bashing?

Q. What's the difference between a commode and a Commodore64?

A. Umm....ahh....hmm....
Arjun

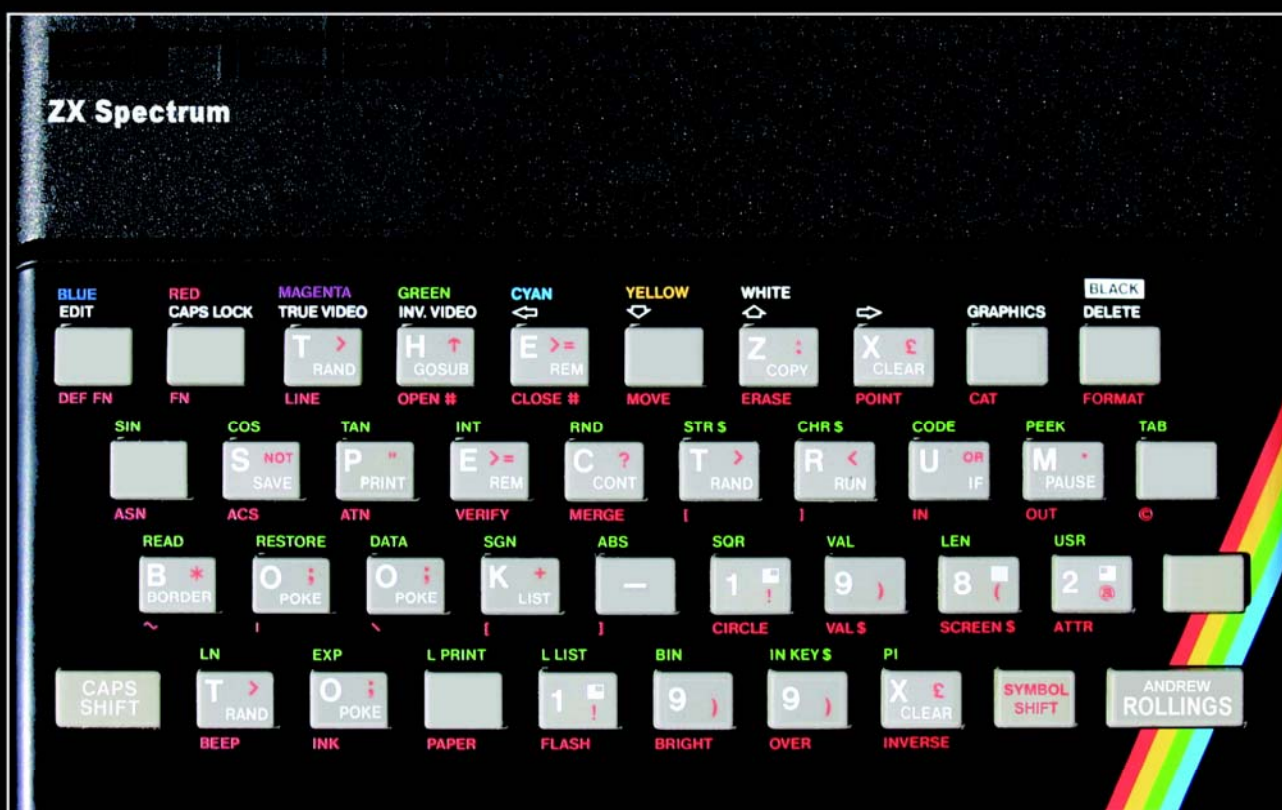
The ZX Spectrum Book 1982–199x

"...for former 'Spec-Chums', this unique tribute to the colour clash of yore will prove hard to resist."

EDGE Magazine

"...We're looking forward to the excellent looking *The ZX Spectrum Book* by Andrew Rollings..."

GAMES™ Magazine



The good news...

- Foreword by **Sir Clive Sinclair**.
- 256 pages of Spectrum games history.
- Available online at <http://www.zxgoldenyears.com> for only £19.99 plus postage.

The bad news...

- Limited print run of 1000 copies. When they're gone, they're gone.

ORDER ONLINE NOW TO RESERVE YOUR COPY

<http://www.zxgoldenyears.com>

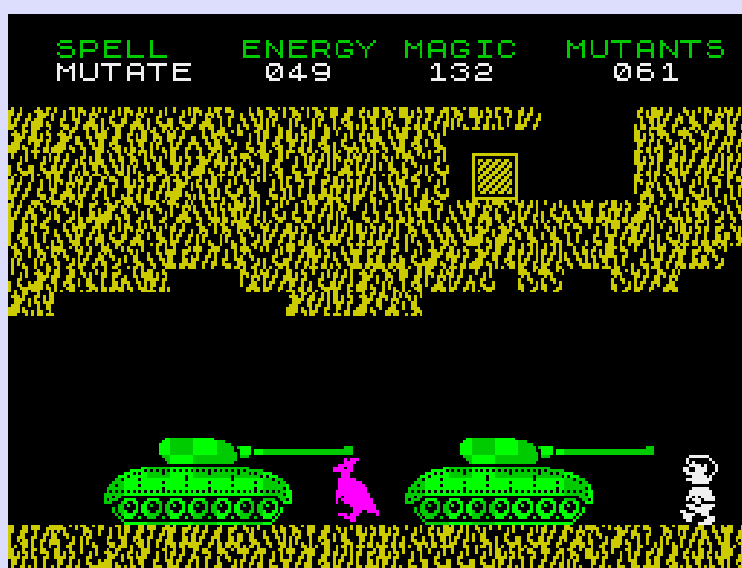
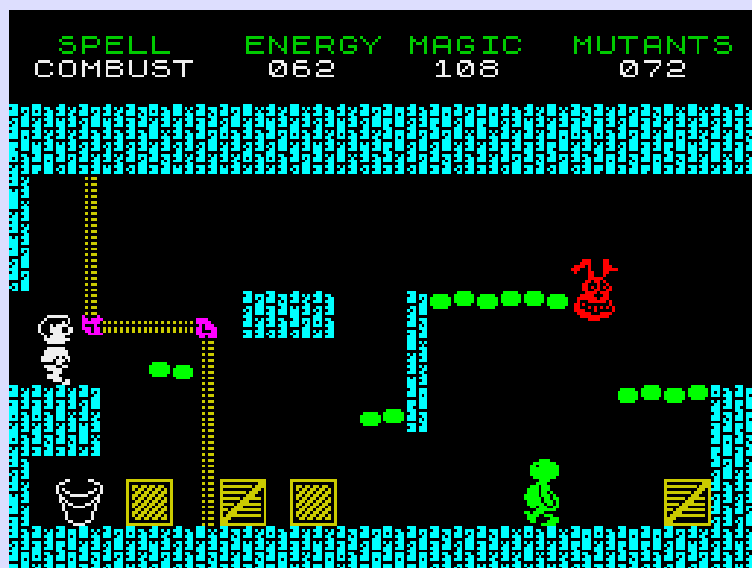
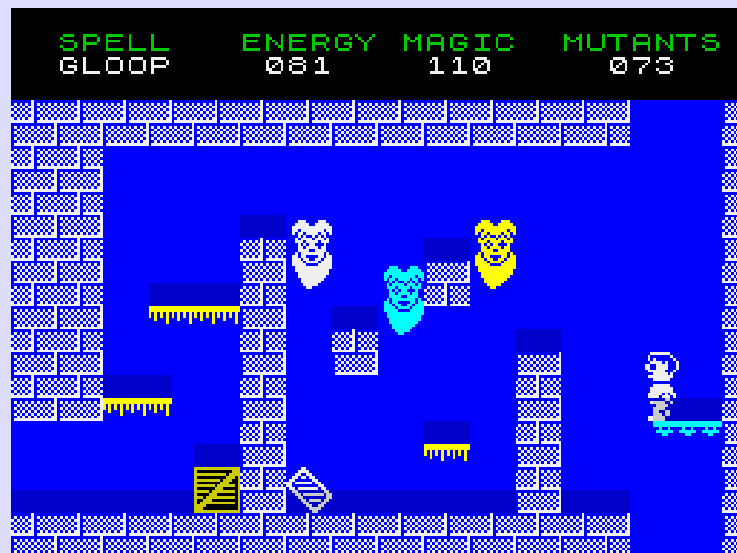


PREVIEW

Fresh on the success of Egghead IV, Jonathan Cauldwell jumps onto his own sequel bandwagon and prepares to save the world from the evil Commodore Fundamentalist Mutants once again with Izzy's third outing...

Having been thwarted in Gloop all those years ago, the Commodore fundamentalist mutants have re-grouped, re-armed and are holed up in a castle hidden deep in the East Midlands. This time they're planning to build a time machine, go back to 1982, and prevent the launch of the ZX Spectrum.

Fortunately, Izzy has used the intervening time to acquire his bronze magic certificate and now has a few magical powers to cast a small number of spells with interesting effects. Each spell requires a different combination of items to be collected before Izzy can perform it. Can he put a stop to their evil plans again?



Another programmer making a great name for himself is Bob Smith. After the release of Farmer Jack, Bob hasn't rested and is currently working on the sequel to the fantastic Stranded. Cunningly titled Stranded 2.5

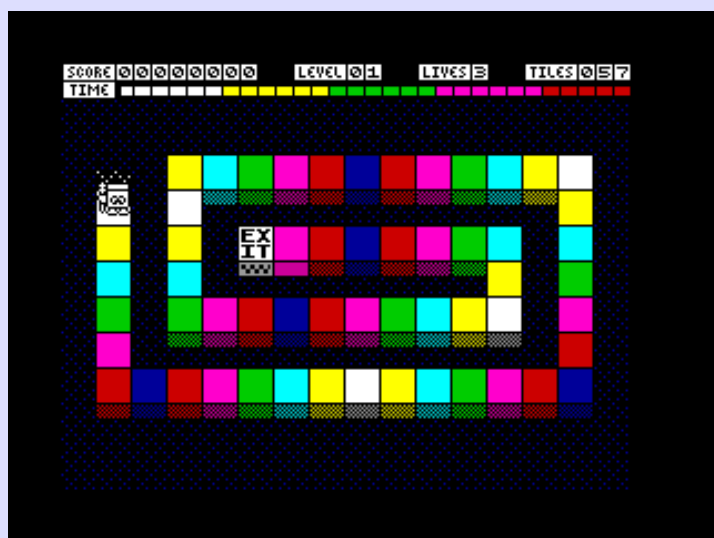
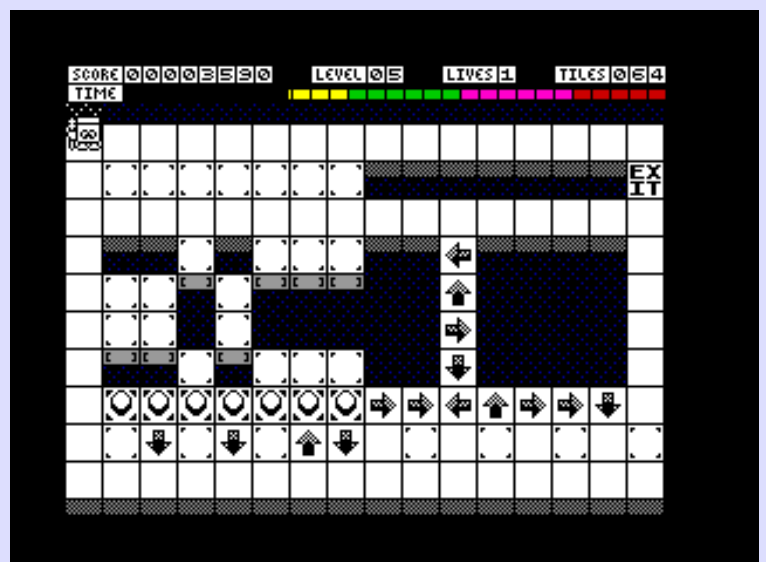
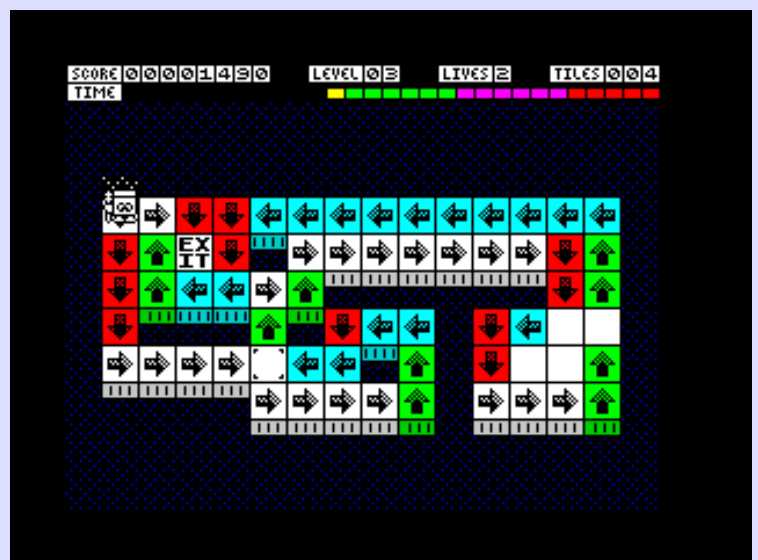
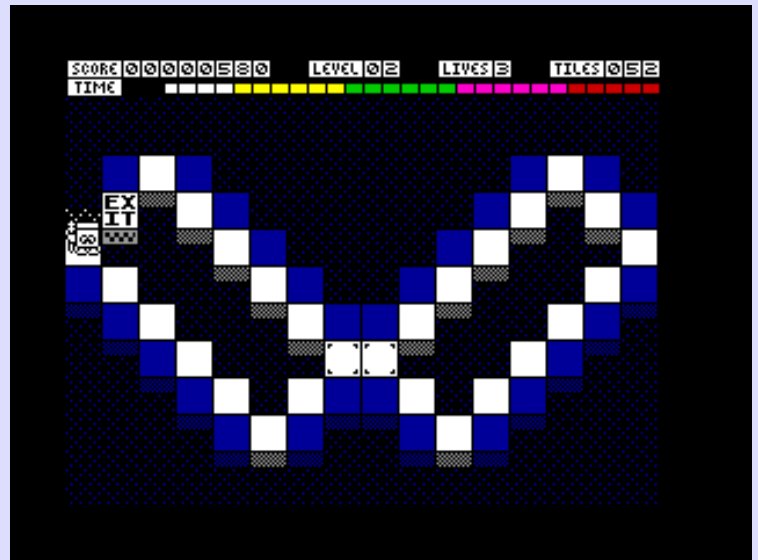
Stranded-2.5 is the kind of game Bob wanted Stranded to be when he first designed it nearly 20 years ago, but didn't have the knowledge or talent to create at the time. So the top-down view of the original has been replaced with by a 2.5D view (hence the title) and the player is now a fully masked character, allowing for the tiles to be more detailed than before - such as having the direction of each conveyor clearly shown on it, and the text on the exit tile.

This doesn't mean it's simply a straight conversion though, as S2.5 has expanded the range of tiles to include conveyors which rotate at different rates, switches which control other tiles, and tiles which must be passed over multiple times before they disappear, to name three. The inclusion of switches allows more puzzles to be included, and the rotating conveyors add a further sense of timing to the levels.

The HUD has also been redesigned to be clearer, with the amount of time left on a level now being plain to see, as are the number of tiles left to be crossed. Indeed, with the HUD being smaller, the game-area has been increased by an extra 2 rows to allow for more complicated levels.

Hopefully Lee du-Caine will be on hand again to provide the 128K music for the game after the fantastic work he did for Farmer Jack..., and that Cronosoft will once again be publishing it.

The story this time around is that Moosh has imprisoned Tysch after breaking the bridge between their worlds and, far from being evil, Tysch's bridge was built to promote trade & friendship. With that now gone, both worlds are suffering, and Tysch must try to escape the prison before he can return home and repair the damage done.

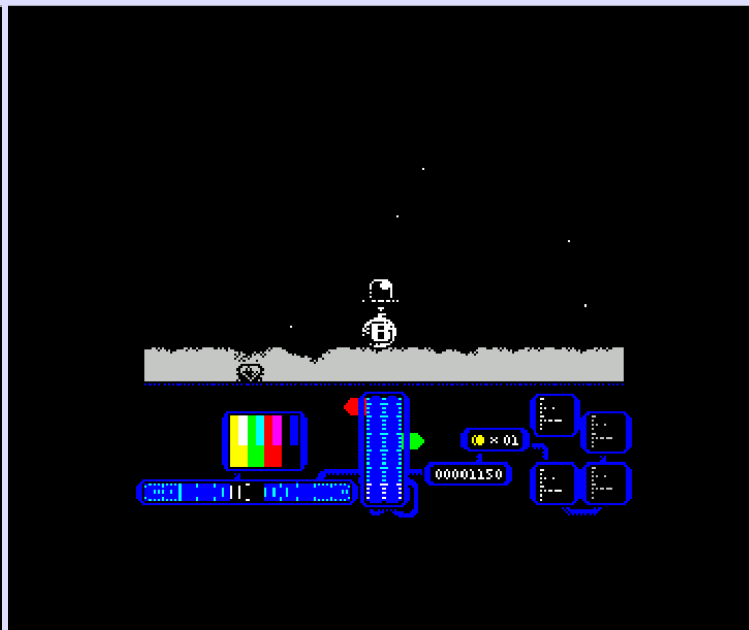


PREVIEW

Who would have believed that after all this time, Joffa Smiff would be back creating games? It seems that he hasn't lost his touch as our preview of *Saucer* shows. It's shaping up to be a classic.

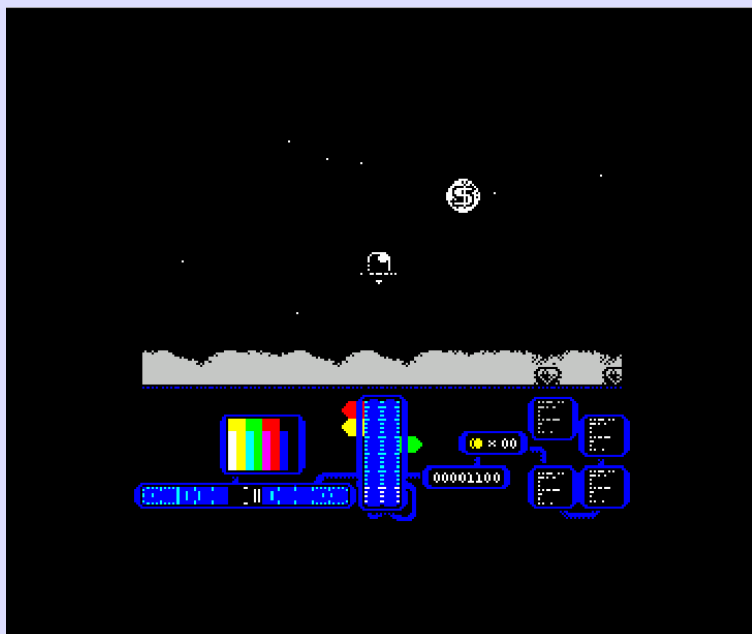


Fly around to shoot the bubbles and collect coins



Drop the bomb to kill all on screen baddies

The aim of the game is simple—stay alive! The game gets harder as time goes on - there aren't levels as such, just a huge landscape. You can fly around the screens shooting the bubbles and collecting coins to boost your power and weapons agility, or jump into your buggy and mine diamonds. You'll be quite surprised at how powerful your little saucer can become. There maybe a few 'big' baddies to kill along the way. The demo plays really smoothly with virtually no slow down with additional bubbles floating around.



Collect the coins to boost your power



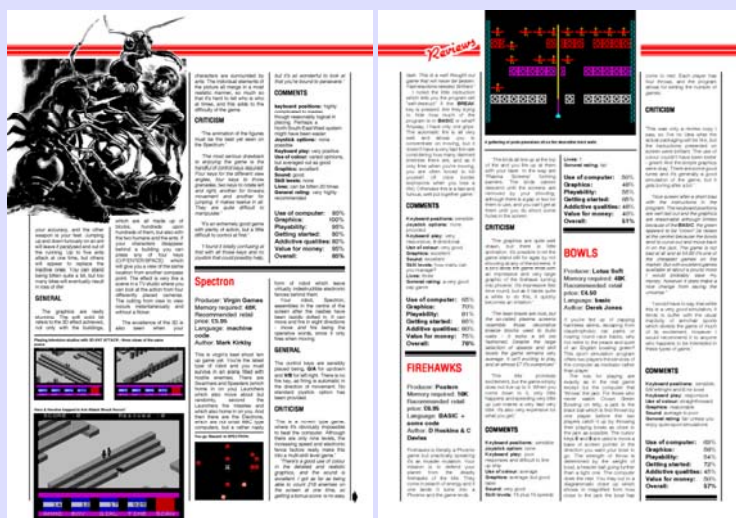
Drilling for diamonds

The demo we played is already very enjoyable. The spot effects are excellent and the game play is simple. It doesn't take long to pick up the rules and find yourself immersed in the game. We look forward to the full review copy to land in our in-box!

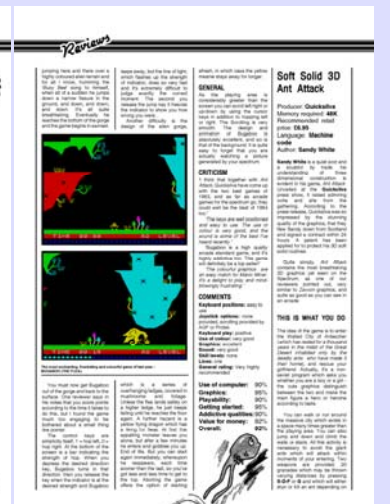
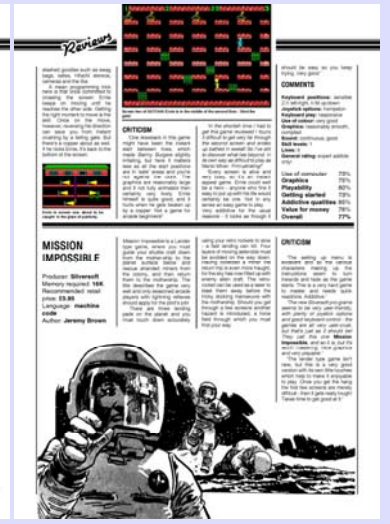
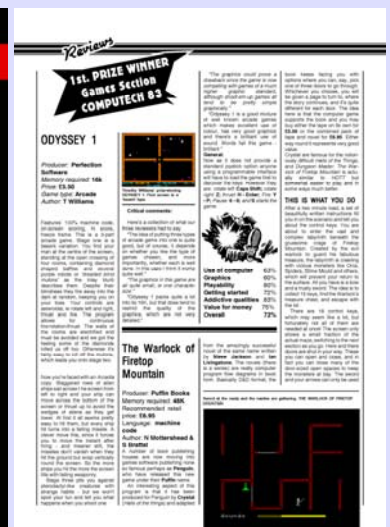
Our final preview isn't a game, it's a book. Paul Dunn has been busy creating a tribute to Crash magazine's reviews.



Looks familiar? The excellent first cover re-created.



Each review features an excellent recreation of the images and screenshots.



Not content with his work on SPIN and BASin, Dunny has been hard at work compiling the entire collection of Crash reviews. The result is a very professional looking collection, with all reviews re-typed with full colour screenshots. One for the collector and a very useful archive. However, there is no release date or even any indication of the format the finished product will take.

DEVELOPER'S DIARY

This issues diary follows Jonathon Cauldwell's journey in the creation of Loco-Bingo.

30.07.2005 - Started work on the new game today, and this time I know how it will work before starting on the code. This one will be a platform game, where the player gets to push trucks around in a manner similar to the Japanese warehousing game, Sokoban. Each truck will have a number on its side, and by pushing a truck off the screen the number on the side is "called" in a bingo game. One by one the player needs to cross off the numbers on a bingo card at the top of the screen; when he has a full house he moves on to the next level. Actually, there's a little more to it than that because by pushing two trucks together the player can combine them into a single truck with a different number on its side.

31.07.2005 - It's been a lot of work, but nearly 400 sprite frames have been drawn over the course of the weekend, though the majority are of a similar design with only minimal changes. Some code is written, mostly graphics routines which are slightly modified versions of the ones used in Egghead 3.

02.08.2005 - The main control routine is in allowing the player to manipulate the train around some platforms and apart from a couple of glitches it's looking and feeling quite nice. I've decided to allow the player to drop off the bottom of the screen in order to re-appear at the top which makes the play area appear slightly bigger than it is.

05.08.2005 - The format of the truck data is decided, and code to display the trucks is written. The trucks will drop from the top of the screen, and the player will need to manipulate them safely around to the bottom avoiding any hazards on the way. They're animating reasonably well, even the little wheels turn as they move left and right, which is a convincing effect. Not a huge amount work is done today though because too much time is spent watching the cricket. If only England were playing badly I'd be able to get some work done, but just for a change the Australians have been on the back foot for a couple of days.

06.08.2005 - Chug - for that is the main character's name now - is pushing those trucks around the screen smoothly, he can stand on them, jump off them and manipulate them with ease. New trucks are dropping from the top of the screen, but after a couple of minutes or so a random truck emerges from nowhere and the game crashes. This bug is proving very difficult to locate, and after a lot of experimentation and debug code has failed to establish the root cause I have to concede defeat for the day. It's midnight, and I'm off to bed.

07.08.2005 - A couple more hours go by and the program just isn't behaving properly. Something very odd is happening, which leads me to suspect there may be a problem with a value being left on the stack somewhere. Sure enough, there is a problem in the player's gravity handling code. Whenever the player stands on a truck an item is left on the stack, leading to unpredictable crashes and other weird and wonderful effects.

09.08.2005 - Most of the sprite handling code is complete, and the player's bingo card is also done. As trucks are pushed off the bottom of the screen any corresponding numbers on the bingo card are flipped over. The routine which chooses the random numbers for the trucks is occasionally getting stuck and really needs reworking when there is time.

12.08.2005 - The eve of the CGEUK, so it'll be an early start in the morning. No work is done on the Spectrum game today, and although progress has been good to date there's still a long way to go. Most of the basic game engine is written and it feels pretty nice to play, as I knew it would. There are a number of issues which are yet to be addressed, such as the time it takes to complete a level which is presently far too long. Extra hazards are planned for later levels, including one or two surprises the player won't be expecting.

13.08.2005 - The day of the CGE arrives, and it seems a little busier than last year. I met and talked to far too many people to list here, but the main memory I'll take away is that of signing so many inlays that a blister developed on my finger, so maybe I was gripping the pen too hard or something. An enjoyable day, though if Matthew Smith turned up I must have missed him. You'd think I spent most of the afternoon at the bar or something.



14.08.2005 - A few more screen blocks are added, including conveyor belts and blocks which will be deadly to touch. Code for these new blocks needs to be written, but can be added later. The truck generator still needs a little work, but it's starting to generate trucks with interesting numbers. One of the tricks up its sleeve is to intentionally generate a new truck which can be combined with one already on screen to give the player one of the numbers on his card. It's then up to the player to spot the combination and push the trucks together.

17.08.2005 - As trucks are pushed off the bottom of the screen the numbers on the side are added to the player's score, providing they exist on the player's bingo card.

Although numbers which do not exist won't add to the score there needs to be some form of penalty to discourage the player from removing unwanted trucks this way. Subtracting the number from the player's score seems a bit harsh as the score is only built up gradually, so another solution is required.

20.08.2005 - Started work on a little sub game which will appear between levels to give the player a bonus. The front end menu screen and high score table routines are also added today, making the game appear close to completion even though it's still a little way off. The whole thing is still silent but until I get around to writing a new AY sound chip driver it's likely to remain that way. A few beeper effects here and there will help, but because all processing stops while they're playing they're of limited use during action sequences.

21.08.2005 - One of the things I noticed at the CGE was the numbers of games which feature attract modes. It's something I've seldom bothered with before, but my games seemed a little boring in comparison as they flipped between menus and high score tables. With this in mind I've knocked up a little demo mode routine to show an in-game screen and move the player's sprite around for a few seconds. It's hardly earth-shattering, but it's slightly less boring than screens full of text.

22.08.2005 - Seven screens are designed, and it's time to start thinking about extra map blocks before any more levels are considered. Conveyor belts would be an interesting feature, and perhaps platforms which disappear and reappear periodically. Work starts on some conveyor belt code, but instead of taking a few minutes to write this soon becomes a problem as things are not working as expected.

24.08.2005 - I've had another idea today, for a character that pops onto the screen and moves around generally getting in the player's way. I might call him the foreman and give him a silly name or something. Patrolling enemies - like the ones in Manic Miner - wouldn't work particularly well. The conveyor belts now have the desired effect, though their images remain static for the time being.

26.08.2005 - The new character has been added for some of the later levels. Starting at the top of the screen, he makes his way left/right across the screen, blowing up any trucks he encounters along the way. It can be a tricky job protecting the trucks you need, and I'm happy enough leaving this saboteur in the game as an extra element to the proceedings. The conveyor belts are moving around, and I can place them around the levels wherever they're needed as the routines are flexible enough to handle any number of them on screen.

28.08.2005 - Some sound effects are in, and for once I've opted to use the Spectrum's 3-channel AY sound chip instead of the beeper. It's actually sounding better than I had hoped.

30.08.2005 - The saboteur has a name - I've decided she is Edna Bucket, an anti-gambling moral campaigner who is desperate to put the online bingo cartel for whom Chug works out of business. She also has no arms and big floppy rabbit ears. Currently Edna pops up after about level 7 or 8, and blows up trucks left right and centre, making the player's task very difficult. Perhaps this needs watering down slightly so that she doesn't appear quite so often, at least not until the later levels.

DEVELOPER'S DIARY



03.09.2005 - Time to forget all about programming Spectrums for the weekend, as I'm off to Skegness with a couple of mates. Once we've dropped the gear off at the guest house it's straight down the pub to watch the England match on the telly, then on to a pub crawl popping in at every arcade we pass until the early hours. It's generally agreed that the lack of decent retro arcade games in Skegness is a bit of a downer but there's not much we can do about that. Still, the evening ends on an amusing note as one of the lads absconds from the nightclub with a ropey-looking young woman from Horncastle. Hmm, Horncastle sounds like a good surname for my Spectrum game's central

07.09.2005 - Just five screens left to design now, as the game edges closer to completion. The high score table checking is in, and I need to decide what sort of name entry routine to write. If there is enough memory left at the end of development I might just write some custom code to do the job. Failing that, I'll use a more conventional routine from an earlier game. Another bug has come to light, which means more trawling through the collision detection code. That's best left for another day though, as I'm off down the pub.

09.09.2005 - A simple little name entry routine is

knocked out in less than an hour, and it's working very nicely indeed. The score has to be converted to the same format as the high score table, as most of the text in the game isn't stored in ASCII, but a code of my own. Work is cut short when Izzy rings up, and I'm off down the pub again. Blimey, you'd think I enjoyed the odd drink or two.

10.09.2005 - The hidden game is pasted in, and fits nigh-on perfectly into the 6K buffer I had to fill. The buffer gets filled up with data as soon as the game proper starts, so the hidden game can only be accessed at the very start. Excluding screen data buffers there are around two to three hundred bytes left to spare, so there isn't room for any music but this can be put to use by adding as many little extra bells and whistles as I can think of.

13.09.2005 - The other day I bought a more up-to-date PC for development, and this one has a working sound card. Once ZX32 is installed I hear the sounds as they appear in the game for the first time, and it's an awful cacophony. Something has gone wrong, and a trawl through the sound data doesn't reveal any problems until I check the sound driver, and realise that the white noise period and looping routines aren't working. Once these are fixed things start to sound the way they were intended. Extra effects will be needed now, as some events don't have any sounds attributed to them.

15.09.2005 - I'm rather pleased with the way everything is going now. The sound effects are done and only four screens are left to design. Edna's visits are a little less frequent and the murderous difficulty settings for later screens have been turned down. There are still a few outstanding bugs, like the player losing a life whenever he pushes a truck onto certain squares, but these are just minor tweaks.

17.09.2005 - The last level layout has been designed and I decide to celebrate with a few cans of lager and a bottle of homebrew "loony-juice". It then becomes apparent that there's a bug in the truck handling code which means they're not always behaving properly on the conveyors. Fixing bugs like this is difficult when the room is spinning, and doubly so when you're using a widescreen television as a temporary monitor.

19.09.2005 - The game is finally complete, the bugs are fixed and we're ready to roll. A BASIC loader program and loading screen need adding and then the emulator tape file can be shipped. However, because The Fantastic Mister Fruity was only released at the CGE last month we'll leave it a few more weeks before releasing this one. That will give me plenty of time to consider my next move...



Hardware, Software, Magazines, Spares and Repairs for the Sam Coupé

Address: Colin Piggot, 16 Belcanto Court, Spalding, Lincolnshire, PE11 3FS

Email: quazar@clara.net

Web: www.samcoupe.com

Picture the scene: It's the 1980s; you're at home having just come back from the computer shop having picked up the latest *Crash Smash*. The tape is running, a fancy title page is on the screen and you can't wait to get stuck in. However, upon loading, all is not well. The game you've bought seems to have playability reminiscent of that type-in you saw in *Sinclair Programs* the other week and you wonder how the magazines could have raved about it? Were they blinded by the spiffy graphics and programming techniques, or just taken in by the hype and the necessity of hanging on to their advertising revenues? Or did you just miss something vital? You sigh, yank out the power cord, and dig out the same game you were playing last week for another bash instead. We relive those moments here.

Airwolf



They said:

Crash: A *Crash Smash*. 90%. "Addictive, playable and fun."

Sinclair User: 3/10. "Best avoided."

YS: 4/5. "One for the masochists."

Matt: My memories of playing this game for the first few times were that I couldn't even get past the second screen for ages: You flew along, this barrier came down, you crashed into it, and you died. After a while I figured you could shoot holes in it, however it regenerated fairly quickly, so you had to do it in a hurry. Once you'd got through you were faced with... another barrier! This was came from left to right and also regenerated rapidly, although you could sneak around it into the next screen and fly back across to get past before it regenerated.

I grimly struggled through the next five screens shooting some square boxes and dodging past gun turrets and radar dishes before giving up in disgust, presuming that I'd never get into the game properly.

Little did I know at the time though that there were only twelve screens to the game and that I'd actually nearly finished it! At last it made sense that the game was so unforgiving: You could play it through in a couple of minutes if you could manage the split second timing needed to avoid all the traps. Upon this news I made one more Herculean (or should that be Masochistic?) effort and finally completed the game after much swearing and thumping of keyboard.

I suppose the creators deserve some credit for the detailed background graphics which still look rather good. It also made me play it to the finish, so perhaps it could even be described as addictive in its own peculiar way? Then again, I shouldn't be too charitable as I never found it a pleasant experience and only really finished the game through some misguided sense of duty.

Despite the cautionary words from *Sinclair User*, the other rave reviews and the TV tie-in brought in the punters and the game sold by the bucket load. A few years down the line it went on to spawn a sequel that was, if anything, even worse. Luckily for us though, none of the magazines gave it anything other than poor reviews.

Black Hawk

They said:

Crash: A *Crash Smash*. 92%. "Addictive, fast arcade game." **Sinclair User:** 7/10. "A shoot-em-up of some complexity."

YS: 7/10. "A pretty good shoot 'em up."

Matt: The trouble with *Black Hawk* was that it was an adaptation of a well established arcade favourite that took a lot of liberties with the format. We'd been playing *Xevious* a lot in the arcades just before this came out and, between marvelling at its furious action and rather splendid – for the time of course – graphics, wondered why



the vertically scrolling shooter hadn't yet made its mark on the Spectrum scene. Then, when we saw the reviews for Black Hawk we thought the revolution was upon us.

On loading the game, the first impression was pretty good. We were treated to a stirring rendition - well as much of one as you'll get in 1 bit mono - of Wagner's Ride of the Valkyries. So far, so good it seemed.

Upon playing the game though, things took a sudden downturn: The graphics were rather drab: OK, the Spectrum isn't cut out for multicolour scrolling shooters and it would be a couple of years before Lightforce brought out the best that it could do, but you appeared to be controlling a small letter U at the bottom of the screen through a landscape of slowly drifting, colour clash-ridden, UDG figures! Also, the game play was a bit on the unconventional side. Rather than having a single screen where you piloted your fighter, the game had two: An attack screen where you could select targets on the ground for a missile strike, and a defence screen where you could have a second shot at taking out the baddies before they killed you, not that it usually helped much to start with.

That meant that, after the ridiculously fast missiles shot down the side of the screen, you were permanently on the defensive and usually got wiped out fairly quickly; which was not exactly forgiving to the beginner.

After persevering with it though, it wasn't really that bad a game. Once you got the knack of taking out those accursed missiles and actually stood a fighting chance of not only taking out some of the ground installations but even getting to the end of the first level. It even had a few nice power-ups you could earn such as an ECM system that would locate hidden radars and the

Wild Weasel that gave you total immunity for a few seconds.

That said, the oddball format didn't exactly make for the most gripping play and before too long it was displaced by superior scrolling shooters. It came back for a budget release just a couple of years later but already looked dated.

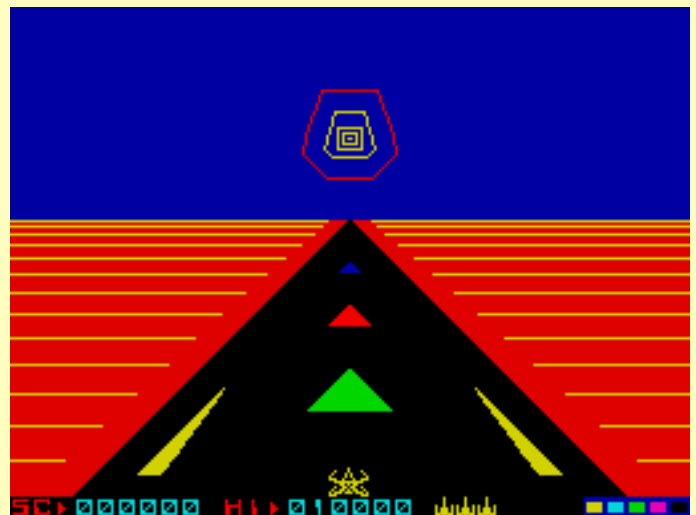
Death Star Interceptor

They said:

Crash: *A Crash Smash*. 92%. "Attractive shoot em up."

Sinclair User: 1/5. "A wholly derivative shoot-em-up."

YS: 2/5. "A whimper, not a bang."



Matt: In 1985 Star Wars clones were all the rage. The arcade game was a big deal and Return of the Jedi was doing the rounds in the cinema. We all wanted to play it at home though on our Spectrum.

The funny thing was though, that before this came out there was already rather good clone of the arcade game in the shape of Realtime's 3D Starstrike and Design Design had also released Dark Star which also played somewhat along the same lines, so it wasn't as if this was even the first entry in the genre.

The game didn't exactly get off to an auspicious start. You had to fly your fighter out of the hangar accompanied by some particularly grotty digitized speech which might have been a good idea on the Commodore 64 version but was a

waste of memory on the Spectrum. Anyway, getting back to the launch sequence, the trouble was that it was exceedingly difficult until you got the knack and it was possible to wipe out your entire fleet of fighters before you even got into contact with the enemy.

I suppose that was just as well as when you got into the shooting part of the game and flew towards the Death Star, it was all too easy. You could just sit in the corner where the enemy fighters would never fly, just taking the odd pot shot at them to relieve the tedium.

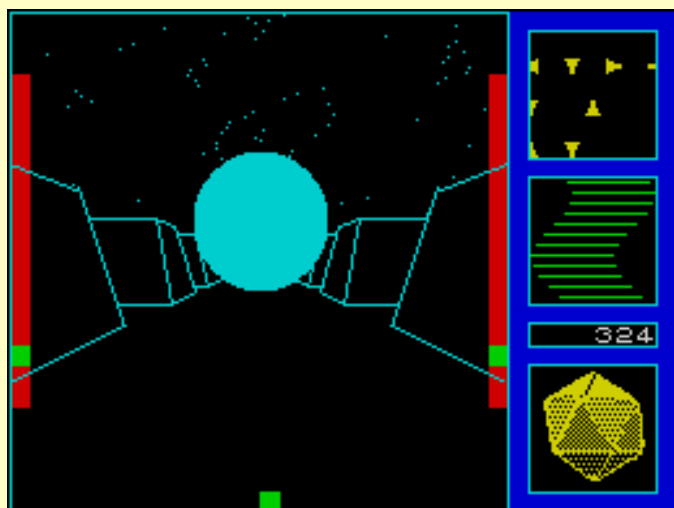
If you fancied more of a challenge you could get out in the middle and mix it up, although that put you at risk from the occasional random fighter bumping into you at the rear.

Eventually, you got to the third stage, which was flying along the bottom of the trench on the Death Star itself.

This was, for my money, the only part of the game that impressed, as you finally had to do fly your spaceship around and shoot things which could get rather frantic. However, you did have to wade through the earlier sections to get there which rather counted against it.

The nail in the coffin for me though was the aforementioned existence of 3D Starstrike and Dark Star, which I promptly went back to playing once I'd swapped away Death Star Interceptor with a friend.

Gyron



They said:

Crash: A *Crash Smash*. 93%. "An astonishing program." **Sinclair User:** A *Classic*. 9/10. "A brilliant game."

YS: 2.5/5. "A 3D maze of considerable complexity."

Matt: Created by Torus, a programming team of four mathematicians who had noble ambitions of revolutionizing the computer games scene, this game arrived with a huge amount of hype. Especially of note was the competition where the first few people to complete the game would get to play off against each other for the prize of a Porsche car. Naturally we all had to rush out and buy it.

Dare I say it though, but the canny YS reviewer had managed to see through the hype and behind all the pretty graphics it really was just a glorified 3D maze game.

The additions of some giant balls that rolled around the maze and towers that would shoot you – both causing instant death – didn't really add much to the game play other than the irritation of having to start again from scratch every time you went near them.

As such, the game play consisted of little more than crawling around the maze, mapping it, finding where you got killed and then trying an alternative route to get past that point. Although making progress was rewarding up to a point, even the promise of the Porsche wasn't enough to keep me interested in the long run.

At the end of the day, the only thing Gyron seemed to excel at was being a demonstration of the Spectrum's capabilities: the 3D view of the maze was spectacular; the 3-colour moving map was a good fiddle to avoid colour clash too; and the rotating icosahedrons in the corner was just icing on the cake.

Torus did indeed go on to create their own little revolution on the Spectrum games scene with their next project, the conversion of the BBC classic Elite, but their first game stands as a monument more to programming skill rather than game design.

POKEPIN

Just when you think the issue is going to be released without any new pokes, along comes Viktor Drozd with a whole shed (ahem) load! So, here's Viktor's very own section!

Firstly, we have a little known Spanish game:- **Akerico** by Crazy Soft and what a good game it is too!

| DESCRIPTION | POKE(s) |
|----------------|---------|
| Infinite Lives | 30427,0 |
| Infinite Time | 24987,0 |

Onto the classic RPG **Bloodwych** by Image Works.

| DESCRIPTION | POKE(s) |
|------------------------|-----------|
| Infinite Hit Points | 52537,126 |
| Infinite Vitality | 56966,126 |
| | 58808,126 |
| Infinite Coins | 62533,183 |
| Open Door With Any Key | 50319,0 |

Lets go and mend some fences in **Colony** by Bulldog

| DESCRIPTION | POKE |
|-------------|----------|
| No Enemies | 55475,24 |

Next we have a nifty Ukrainian game - **Crime Santa Clause: Déjà vu** from BrokImSoft

| DESCRIPTION | POKE |
|-------------|-----------|
| Immunity | 31842,201 |

And so onto **Egghead 4: Egghead Entertains** by Jonathan Cauldwell

| DESCRIPTION | POKES |
|----------------|-----------|
| Infinite Lives | 37238,182 |
| Immunity | 37716,201 |
| Infinite Time | 32600,182 |

Boulder dash fans are in for a real treat now – an immense amount of pokes for one of the best set of games ever! (In the editor's opinion). These pokes are in addition to the ones in the tipshop...

Boulder Dash was re-released by Prism Leisure Corporation. These pokes will only work with that version.

| DESCRIPTION | POKE(s) |
|---------------------------|-----------|
| Infinite Lives | 31018,182 |
| Infinite Time | 26021,33 |
| Immune To Boulders | 26927,195 |
| Boulders Don't Fall Down | 26842,195 |
| Boulders Don't Roll Down | 26827,201 |
| Immune To Diamonds | 25475,195 |
| Diamonds Don't Fall Down | 25607,195 |
| Diamonds Don't Roll Down | 25592,201 |
| Immune To Nasties | 25212,195 |
| Immune To Diamond Nasties | 25202,201 |

Boulder Dash 2: Rockford's Riot - Prism Leisure Corporation PLC

| DESCRIPTION | POKE(s) |
|---------------------------|-----------|
| Infinite Time | 26028,33 |
| Immune To Boulders | 26963,195 |
| Boulders Don't Fall Down | 26878,195 |
| Boulders Don't Roll Down | 26863,201 |
| Immune To Diamonds | 25482,195 |
| Diamonds Don't Fall Down | 25614,195 |
| Diamonds Don't Roll Down | 25599,201 |
| Immune To Nasties | 25148,195 |
| Immune To Diamond Nasties | 25138,201 |

Boulder Dash 3 -Prism Leisure Corporation PLC

| DESCRIPTION | POKE(s) |
|---------------------------|-----------|
| Infinite Lives | 31008,182 |
| Infinite Time | 26011,33 |
| Immune To Boulders | 26917,195 |
| Boulders Don't Fall Down | 26832,195 |
| Boulders Don't Roll Down | 26817,201 |
| Immune To Diamonds | 25465,195 |
| Diamonds Don't Fall Down | 25597,195 |
| Diamonds Don't Roll Down | 25582,201 |
| Immune To Nasties | 25202,195 |
| Immune To Diamond Nasties | 25192,201 |

INTERVIEW



It's not every day that someone emails you and says 'I just interviewed Kevin Toms'. But that is just what happened.

So ZX Shed is proud to present you with the EPL Talk interview conducted by The Gaffer. We will find out what makes Kevin tick and hear about his new project - footymax.

TG - what's your best guess at the amount of computer games you've sold in your career?

KT - it's a couple of million, I'm not sure exactly. 2-3 million probably.

TG - when you created football manager back in 1982, how revolutionary was it at the time?

KT - well it was, we didn't really realize at the time, but it was, I had been working on stuff like that before, and really what happened was it came together with home computers and the ability to create that kind of game and messed around with it as a board game and a rudimentary computer game and then with the advent of the home micro, put it all together and took advantage of the game intelligence that they provided. A lot of the games at the time were really just clones of space invaders or very simple games, so a strategy game as relatively complex as that was quite different at the time. I also did some other games in-between as well. I did a game called software star and a game called president which was just not as well known because football is such a big sport and therefore just got more attention basically but they were other strategy games on other subjects. One a hits game and one about running a country.

"A lot of the games at the time were really just clones of space invaders or very simple games, so a strategy game as relatively complex as that was quite different at the time."

TG - one of the things that's very common in football manager any your new endeavour which is footymax is the simplicity of the games, the way it's easy to use, easy to understand and easy to play. Is that something that's intentional?

KT - absolutely intentional. There is a style to what I do and one of the things that I like to do is to make it accessible and apparently straight forward, but the subtlety is underneath and so it needs two things. One it needs a straight forward interface and also have enough variety in it so that although it appears straight forward it isn't simple and therefore boring and there is a lot of different things that can happen when you play. But accessible and not too much clutter is important to me and I try to keep to

that.

TG - can you tell us more about footymax?

KT - really what I'm trying to do there is a similar approach to what I did on the home computers but on the internet and take advantage of all the things that the internet provides so it is a football management game, it is similar in concept to the original football manager but with game play it is updated and I guess one of the important things is that although it is on the internet its not one of those games where you download it, its very much a game that you play just in the standard browser. It was important to me that you can play it from wherever and didn't have to have anything special on your machine. You can go into an internet cafe to play - whatever you wanted to do. Its very easily available to you and to give you the same feel and interactive pleasure of a typical PC football management game like the sort of thing I did before but to be on the internet, whereas some of the other football management games on the internet are kind of like play by mail where you make your move & wait to see what happens a week later. This you can just play & play & play just like PC games and also there's an increasing amount of taking advantage that you can play against other people. And that's something I had a lot of plans of for the time.

TG - so the way the game works, basically you can start off as managing a country and then you can go in and edit down change it to managing club teams if you like?

KT - yes. It's got a lot of customizing in what you can do, so you can change it and make it the way you want it to be. It starts of with this concept of playing with national teams and the league which was a bit of fun that I thought of but actually a lot of people seemed to like it and I quite enjoyed myself to play as England against Rwanda or whoever it happens to be - seems to be a fun concept and I guess the fact that it came out around the time of the world cup kind of fitted in as well. But yeah, you can make it the premier league or whatever you want it to be.

TG - The logic that goes into the computer program, is there any similarities between football manager and footymax, or is it completely different?

KT - I think the three football games before this one were quite different, so I guess no. In terms of it coming from me and the way that I think and design, yes, but the algorithms, I don't need to use anything that I've done before, except it's just my own way, my style, so completely fresh and creative.

TG - Why a footy game now? And why not something else?

KT - I love football and that's one of the main reasons. It's something I get a lot of entertainment out of and I've studied a lot and watched a lot so that has to be the main reason. Secondly, at this stage I think that there's a huge amount of scope for what I can do with this game. I don't see any limitations in where I can take it and the things I can add to it and how it can stand, so it'll keep me going for a fair while yet work with just football and it's just such a good thing to work with and it's popular as well so lots of people can enjoy it.

TG - Can you tell us a little bit about the team that you support and some of your football memories while growing up in England?

KT - I'm very much one of those that comes from a small town in England - therefore I grew up on low division football, which makes me very tolerant of football really because I wasn't lucky enough to live near a premier league team or any top division teams. It was Torquay united that I followed which sometimes barely hang on to their status in the English league at all.

TG - Do you think being a Torquay supporter had any influence on your games? For example with football manager and with footymax, both of them you start at the bottom division and you have to work your way up through the divisions to get to the top. Was that something that was intentional?

KT - It probably had some impact as I was used to that. I think that also it provided a challenge to start there and build up. It's certainly a concept in my mind and I think I've carried that through to footy max as well. You start low & try to build yourself up and climb up and that's fun - there's a lot to that, trying to build something out of something small.

TG - How does it work with footymax? Is it a monthly subscription fee?

KT - Yes that's right - it is a monthly subscription. You can play as much as you like which you pay on a monthly basis or you can subscribe for three months, six months or a year for discount prices.

TG - Do you think there are some advantages to

having this game online for you as apposed to the old days where you would have take a game and put it onto disk, then once it was on disk there's no turning back?

KT - absolutely - that's a key part of what I am doing. That was one of the key attractions of creating it this way. Whereas before, I would put out 50,000 CD's and they've gone out there and I've thought of some great idea to add a few weeks later, there's no chance of me doing that unless I was able to recall all those CD's back, so this is much more attractive and already I've had a couple of minor adjustments and its only been out about a month now, and every few months there will be updates for it. There's something in the pipeline. I've built an infrastructure that I can enhance the game upon, so one thing I can certainly see is that anyone who subscribes is joining an evolving game, not just what it is now, but what it's going to be in the future. And certainly to be very responsive to peoples feedback as well.

"It was Torquay United that I followed which sometimes barely hang on to their status in the English league at all."

TG - What's your opinion of the other type of soccer video games out there such as the FIFA series and the pro evolution series?

KT - they all have their place and I play them from time to time. Some of them I think over do it a bit on controls and get a bit hard work to learn. It's already for people that really dig into it, but not everyone wants to spend so much time learning to play. Its horses for courses - sometimes you want to play stuff like that and sometimes you want to think and take your time - have a more relaxed game. One of the things about the football management games and this kind of genre is that you can play for longer because it's not such an effort to play. You can pace it and make it more long term. It's always been true - playing joystick games for hours and hours is harder than playing a game like this.

TG - At the height of your popularity in England, what was it like living there. Did you have strangers coming up to you and talking to you?

KT - Yes it did happen. It happened in all sorts of places like petrol stations or on holiday. That was very much part of it. I used to think it was a bit like being a science fiction author or something - famous in a narrow area. It was fun, I enjoyed it. I enjoy dealing with the public anyway so it suits me.

TG - What about football celebrities? Did you ever get to meet any of those from working in the computer games industry, but working on football titles?

INTERVIEW

KT - sometimes. I need to think back now... With some of the football management games, but not too much now.

TG - What about Lawrie McNemeny, the Southampton manager from the 80's?

KT - that's right. There were a couple of others. Yeah occasionally.

TG - You live in New Zealand. What's it like following the English premiership there? And the world cup. How does that work out in terms of time zones etc..?

KT - Yeah that can be a problem and the world cup for example, it was fortunate in that most of the matches there were starting at 7am, so you get used to watching in the morning. The worst part was watching England at 3am, which I got up a couple of times to do, so it's not ideal in that way. For the premier league, you've got Sky TV, so you can see the premier league fine. Probably the hardest thing for me is the fact that I used to watch Torquay United and the local football hero's. There's no premier league type football here that you can watch live, but I have a tolerance there having watched low division football so it doesn't bother me much. I still enjoy it.

"The worst part was watching England at 3am, which I got up a couple of times to do."

TG - can you tell us a little bit about the football leagues in New Zealand?

KT - the main league here is just 8 teams so its quite small. The population of New Zealand in total is only about 4 million which is the size of one city in England in one country.

TG - and what about watching the premiership in England. What's the experience like down under?

KT - there are a lot of ex-pats here, a lot of British people as well and a lot of Europeans so there's quite a lot of people that follow football here. It's very popular.

TG - what's it like knowing that your games have left a permanent imprint on the minds of millions of soccer fans worldwide?

KT - generally it's good. I get emails from people telling me about their past experiences which are very good. It's pleasing to have had that impact. I'm very happy with it. I'm hoping to repeat it with what I am doing now.

TG - do you see any parallels between the football manager of 1982 and the footymax of 2006?

KT - the parallel for me would be that the scope of what is to be done yet with the internet. That would be the main parallel. I think there's an awful

lot to do and an awful lot to exploit yet so the parallel is a technology that's not being fully utilised yet for this kind of product.

TG - How does footymax differ to football manager when you were making it in the 80's?

KT - the main thing I see that's different this time is the mutli user stuff. I've deliberately tried to create a game that you can play like a pc game so you don't have to wait for other people to be online to play, but you are competing with other people. It's really that side of it. The football games I have produced before, you were playing in isolation. I very much want to exploit the fact that you're not in isolation now, you're online with other people, but at the same time, that's the catch-22 because at the same time, I wanted to not be limited by the fact that not everybody will be online at the same time. It's an interesting area, but also an area that has a lot of scope.

TG - what would you say is the most surprising thing about you that most of your fans wouldn't know?

KT - my sense of humour. I don't think that's obvious. Maybe it shows through sometimes, I don't know. I like a good laugh & a joke.

TG - Is that something you've included in some of your games? Maybe Easter eggs or some other pieces of humour? Or is that more outside of the computer game industry?

KT - There is a certain amount of seriousness in the games themselves and for example I like it to be so that I can't cheat - I don't know any way of beating the game myself, so I am on a level playing field with everyone else, so I don't like to put back doors in to change things. I don't want to have any advantage over anyone else - it's a bit purist of me I think but it's the way I want to do it. It should be entertaining for me as well. So I'm serious in that way about how I do that. Of course, the whole thing in the end has got to be fun. That's the whole purpose to it. That's one of the reasons I try to make it straight forward and easy to use so it's not hard work to learn to play.

TG - I've been playing the footymax game online for the last couple of weeks and one of the things I've been wondering with footymax & football manager is when you are playing the games is there a logic or secret formula that I can try to figure out in order to determine how to beat this game? Or is it something that's randomly generated?

KT - part of my job as a games designer is to make it so that you can't do that. The other thing I try to do is to understand that people play in different ways and develop different styles and I like to try to provide the tools for different styles of play to be

available so one person can play one way and they will tell you that this is the way to be successful and another person will come up with something quite different and say this way works, and it should be that that's true - the different ways will work and it should be that there is enough variability there so that you can't come up with anything that guarantees success every time. It should be that you can be successful but there is no key that unlocks it all.

TG - which I guess in a way typifies the game of football in that football manager and players can control a lot of things but when you go out on the pitch and play for 90 minutes, a lot of things can happen and you never know what's going to happen.

KT - that's right. And that's what I was saying earlier on in that it seems simple but its subtle underneath with the complexity underneath and that's what keeps enough variable in there that you can't get that absolute certainty or follow steps to a guaranteed win. Skills are very important and have got to count but not a simple formula like that.

"Inventing games seemed to be a habit of mine, and I was doing that from quite young so I was really putting the two things together."

TG - when you were growing up in England, were you more of a football fan or more of a computer fan, or more of a balance between the two?

KT - definitely a balance between the two. I used to be into games writing before I got hold of computers. Making games like board games or ball games. Inventing games seemed to be a habit of mine, and I was doing that from quite young so I was really putting the two things together. Football has always been the sport that I love more than any other by a long way. I watch others and enjoy them but its football that I really enjoy.

TG - being somebody that has a creative and inventive mind with regards to games, when you look at games like football, do you look at ways to change it or things that could make it better?

KT - over the years when I've seen rule changes, they've been pretty good. I thought of one during the world cup. The one I would do something about now is to deal with time wasting. And the substitution one. The way people use substitution to waste time at the end of the match which is irritating. I think in the end, it's a sport, but its also entertainment, the people are paying to watch it and you've got to think about them. The two things the managers put substitutes on to spoil just to lose time at the end of the match for the last minute or so when they know it will just use up time. And the other one is the players being able to time waste as

well like taking the ball into the corner. I guess the thing is that part of it that it's down to the referee to do the timing and they try and manipulate the referee and sometimes it could be better if someone else was timing the match as in other sports. But they clearly made the decision to give the referee control because it's an arbitrary thing, they can decide this teams messing around so I can add a couple of minutes.

TG - one of the things we just saw in the world cup was a lot of diving. What about that?

KT - ah - the diving one! I think that one thing I would go for is post match bookings and sending off. If you're caught cheating and the referee didn't spot it, they can still book you and send you off afterwards.

TG - looking back at your life so far, what would you say has been your greatest football memory so far?

KT - that's a really good question. There are so many things, recent things like the last world cup, seeing some of the skills being displayed. The last one that comes to mind is Zedan's penalty. That chip penalty, the whole trick of doing that was superb. Argentina's passing in their 6-0 win, incredible. I can't think of just one event, just the display of skill that's great to watch, its one of the things that makes football so good.

TG - last question. What else can you tell us about footymax that we haven't covered already?

KT - I guess it's different because a lot of games like that so far have been play by mail type games - you enter your moves and come back later. I'm trying to give people the PC experience online as well as being multi-user. I think the other thing about it is that it's an evolving game and I really want people coming into it knowing that its something that's going to grow and get better and better.

TG - thanks again for the interview. I'm looking forward to playing the soccer simulation game.

KT - you're still calling it soccer though!

TG - yes, it's tough for a lot of Americans living in America with the whole thing around American football and our football.

KT - that's the thing that FIFA are working on. I believe there are three countries in the world that use the word soccer - New Zealand, Australia and the US. In New Zealand and Australia they have switched to football now which just leaves the US using the word soccer. That's a bigger thing to crack though.

You can listen to more pod casts on the EPL Talk website at <http://www.epltalk.com/> And of course, you can play footymax at <http://www.footymax.com/>

Reading The Keyboard.

The first thing you want to do when learning to code is to display something on the screen. The second thing you want to do is move it. Once you got it on the screen and moving it about comes the next stage... to control it.

So how do we do that? Well this article will focus on the most available device to the Spectrum programmer, the keyboard.

Essentially there are two ways to read the keyboard; the first is the ROM routines readily available to the Spectrum machine code programmer. The second is to access the keyboard directly by writing your own routines. The latter is obviously much faster than the former as you are not doing all the checks the ROM routines have to do to check all the various combinations the system expects.

The ROM routines.

In the System Variables there is a location called LAST_K, its address is \$5C08. It does exactly what it says and holds the last key pressed. This location is updated every 50th of a second (or every 60th of a second in the USA). Because the key is in its ASCII format this makes life very easy to display to the screen.

One thing you must do first though it to clear LAST_K before you read from it, this will ensure that you correctly read the keypress.

Here's a simple piece of code to read the keyboard using LAST_K and then display it on the screen.

```
LAST_K      EQU    $5C08          ; LAST_K is System
                                   Variables location
                                   $5C08

BIG_LOOP    LD      A,0
                                   ; Put zero into
                                   LAST_K
                                   ;
LOOP        LD      A,(LAST_K)    ; Read LAST_K
                                   ;
CP          0                    ; Check if a key
                                   pressed
                                   ;
JR          Z,LOOP              ; Keep checking
                                   until a key is
                                   pressed
                                   ;
RST         $10                 ; PRINT it to the
                                   screen
                                   ;
JR          BIG_LOOP
```

Or if you are coding a game instead of the RST \$10 you could have the following.

```
CP          'o'                 ; Check to see if 'o' has
                                   been pressed
CALL        Z,MOVE_LEFT         ; IF so execute MOVE_LEFT
                                   routine
CP          'p'                 ; Check to see if 'p' has
                                   been pressed
CALL        A,(LAST_K)          ; Read LAST_K
CP          Z,MOVE_RIGHT        ; If so execute MOVE_RIGHT
                                   routine
```

This is probably the easiest way to read the keyboard but is probably also the most inefficient; it is slow and crude but ideal for the machine code newcomer. This routine would be ideal for some sort of word processing program.

There is a neat routine called KEY-SCAN at \$028E which returns two values in D and E. If no key is pressed both D and E return \$FF. If just one key is pressed then D returns \$FF and E returns the value for that key. Now things get more interesting, if you press the CAPS SHIFT key with and other key then D will return 27 and E will return the value of the other key pressed. Similarly with SYMBOL SHIFT D will return 18. If CAPS SHIFT and SYMBOL SHIFT are pressed together then D will return 27 and E will return 18. It does have limitations, it cannot read 2 or more keys pressed simultaneously (other than CAPS SHIFT and SYMBOL SHIFT) but is definitely useful if all you need to do is detect a pressed key without any great need for speed.

This diagram shows the value for each key.

Fig. 1

| | | | | | | | | | |
|---------------|----|----|----|----|----|----|----|--------------|-------|
| 24 | 1C | 14 | 0C | 04 | 03 | 0B | 13 | 1B | 23 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| 25 | 1D | 15 | 0D | 05 | 02 | 0A | 12 | 1A | 22 |
| Q | W | E | R | T | Y | U | I | O | P |
| 26 | 1E | 16 | 0E | 06 | 01 | 09 | 11 | 19 | 21 |
| A | S | D | F | G | H | J | K | L | ENTER |
| 27 | 1F | 17 | 0F | 07 | 00 | 08 | 10 | 18 | 20 |
| CAPS SHIFT | Z | X | C | V | B | N | M | SYM SHIFT | SPACE |

The major problem with KEY-SCAN is that it corrupts all the registers so before you call it you will need to stack any registers you are currently using. Let's continue with our assumption that we are writing a game.

```
KEY-SCAN    EQU            $028E    ; KEY-SCAN
                                   is at ROM
                                   location
                                   $028E

LOOP        PUSH            BC        ; Stack BC
                                   register
                                   ;
                                   PUSH    HL    ; Stack HL
                                   register
```



```

CALL    KEY-SCAN      ; Call ROM
                        routine
POP      HL            ; Restore HL
                        pressed
POP      BC            ; Restore BC
LD       A,E           ; Get the key
                        code into A
CP       $25           ; Check 'Q'
                        pressed
CALL    Z ,MOVE_UP     ; If so, execute
                        MOVE_UP routine
CP       $26           ; Check if 'A'
                        pressed
CALL    Z ,MOVE_DOWN   ; If so, execute
                        MOVE_DOWN
                        routine
JR      LOOP

```

This will continually check for key presses and perform the appropriate routine upon the correct key being pressed.

Directly Accessing The Keyboard.

When you've got a dozen sprites speeding around the screen in a 3D universe with parallel scrolling then a ROM routine to get the keyboard input is just not gonna cut it!!! We need to get our hands dirty and access the keyboard directly.

Firstly we'll get the technical stuff out of the way and examine how the Spectrum reads the keyboard.

If you look at the keyboard it is basically set out as 4 rows of ten as indicated previously in Fig.1 but the Spectrum looks at it a lot differently. It regards the keyboard as 8 half-rows of 5 keys. The Spectrum is connected to the keyboard via 2 ribbon cables, one has 8 tracks the other has 5. On the larger cable each of the tracks are connected to the hi-byte of the address bus (bits 8 to 15) whilst the smaller cable is connected to the lowest 5 bits (0 to 4) of the data bus.

Every 50th of a second the Spectrum scans the keyboard, it does this by applying a current to each of the address lines in turn. Each of the 5 keys in their corresponding half-row is a 'switch' and connects each of the data lines with the address line and allows a current to flow when depressed. The Spectrum reads each of the 5 data lines and if a current is detected on a line the Spectrum then knows which key was depressed.

The address lines are labelled A8 to A15 and the data lines D0 to D4. The diagram in (Fig. 2) explains this more clearly.

Fig. 2

| A11 | | | | | A12 | | | | |
|-----------------------|----------|----------|----------|----------|----------|----------|----------|----------------------|--------------|
| D0 | D1 | D2 | D3 | D4 | D4 | D3 | D2 | D1 | D0 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| A10 | | | | | A13 | | | | |
| D0 | D1 | D2 | D3 | D4 | D4 | D3 | D2 | D1 | D0 |
| Q | W | E | R | T | Y | U | I | O | P |
| A9 | | | | | A14 | | | | |
| D0 | D1 | D2 | D3 | D4 | D4 | D3 | D2 | D1 | D0 |
| A | S | D | F | G | H | J | K | L | ENTER |
| A8 | | | | | A15 | | | | |
| D0 | D1 | D2 | D3 | D4 | D4 | D3 | D2 | D1 | D0 |
| CAPS SHIFT | Z | X | C | V | B | N | M | SYM SHIFT | SPACE |

The keyboard is selected by sending address line A0 low, therefore the low byte of our input port address will be \$FE (%1111110). We use one of two instructions, either

```
IN A, ($FE)
```

or we load the C register with \$FE and use the Instruction

```
IN r, (C)
```

Where r is a single register. Before we can do that though we must load the hi-byte of the address into A (if we are using IN A,(\$FE) or B (if we are using IN r,(C)).

So to read the half row 1 – 5 we could

```
LD A,$F7
IN A, ($FE)
```

Now suppose we were looking to see if 'Q' had been pressed.

```

LD      A,$FB          ; Half row Q - T
IN      A, ($FE)       ; Get the keyboard input
AND     1              ; Test for bit 0
CALL    Z ,MOVE_UP     ; If pressed CALL
                        routine.

```

See, easy eh?

That's it really in its simplest terms. You now have all you need ensure your super fast sprite whizzing mega game has the keyboard input to keep up with it.

One thing to bear in mind is that testing bits is not the only way to detect a keypress. We can also use the AND instruction. The other difference is if a key

has been pressed the Z Flag will be reset so we need to use NZ rather Z for our jump condition.

BIT 0,A = AND \$01
 BIT 1,A = AND \$02
 BIT 2,A = AND \$04
 BIT 3,A = AND \$08
 BIT 4,A = AND \$10

The previous piece of code will now look like this.

```
LD      A,$FB      ; Half row Q - T
IN      A,($FE)     ; Get the keyboard input
BIT     0,A         ; Test for bit 0
CALL    Z ,MOVE_UP  ; if pressed CALL routine.
```

Fig. 3 explains which address lines to read and which data lines to test.

Fig. 3

| F7 | | | | | EF | | | | |
|-------------------|----------|----------|----------|----------|----------|----------|----------|------------------|--------------|
| 0 | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 0 |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 0 |
| FB | | | | | DF | | | | |
| 0 | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 0 |
| Q | W | E | R | T | Y | U | I | O | P |
| FD | | | | | BF | | | | |
| 0 | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 0 |
| A | S | D | F | G | H | J | K | L | ENTER |
| FE | | | | | 7F | | | | |
| 0 | 1 | 2 | 3 | 4 | 4 | 3 | 2 | 1 | 0 |
| CAPS SHIFT | Z | X | C | V | B | N | M | SYM SHIFT | SPACE |

Let's now look at a couple of examples to see how the keyboard is read in professional games.

Hungry Horace

Hungry Horace is a simple game with just 4 controls, the direction. As you can see from the listing the method used is very straight-forward and easy to read and understand. Not much really needs to be added to as if you have understood so far then this should be very clear. One thing to mention is the testing for keys 1 to 4; this would be for early joysticks.

```
62C4 3EFB      LD      A,$FB      ; Port Address for Q - T
62C6 DBFE      IN      A,($FE)
62C8 E601      AND     $01        ; Test for 'Q' - UP
62CA 2808      JR      Z,$62D4
```

```
62CC 3EF7      LD      A,$F7      ; Port Address for 1 - 5
62CE DBFE      IN      A,($FE)
62D0 E608      AND     $08        ; Test for '4'
62D2 2007      JR      NZ,$62DB   ; Jump to next keyread
62D4 3E00      LD      A,$00      ; 0 means Horace moves up
62D6 326F7C    LD      ($7C6F),A
62D9 1843      JR      $631E
62DB 3EFE      LD      A,$FE      ; Port Address for CS - V
62DD DBFE      IN      A,($FE)
62DF E602      AND     $02        ; Test for 'Z' - DOWN
62E1 2808      JR      Z,$62EB   ; Jump to move routine
62E3 3EF7      LD      A,$F7      ; Port Address for 1 - 5
62E5 DBFE      IN      A,($FE)
62E7 E604      AND     $04        ; Test for '2'
62E9 2007      JR      NZ,$62F2   ; Jump to next keyread
62EB 3E02      LD      A,$02      ; 2 means Horace moves left
62ED 326F7C    LD      ($7C6F),A
62F0 182C      JR      $631E
62F2 3EDF      LD      A,$DF      ; Port Address for Y - P
62F4 DBFE      IN      A,($FE)
62F6 CB47      BIT     0,A        ; Test for 'P' - RIGHT
62F8 2808      JR      Z,$6302   ; Jump to move routine
62FA 3EF7      LD      A,$F7      ; Port Address for 1 - 5
62FC DBFE      IN      A,($FE)
62FE E602      AND     $02        ; Test for '2'
6300 2007      JR      NZ,$6309   ; Jump to next keyread
6302 3E01      LD      A,$01      ; 1 means Horace moves right
6304 326F7C    LD      ($7C6F),A
6307 1815      JR      $631E
6309 3EDF      LD      A,$DF      ; Port Address for Y - P
630B DBFE      IN      A,($FE)
630D CB57      BIT     2,A        ; Test for 'I' - LEFT
630F 2808      JR      Z,$6319
6311 3EF7      LD      A,$F7      ; Port Address for 1 - 5
6313 DBFE      IN      A,($FE)
6315 E601      AND     $01        ; Test for '1'
6317 2014      JR      NZ,$632D
6319 3E03      LD      A,$03      ; 3 means Horace moves left
631B 326F7C    LD      ($7C6F),A
```

| | | | | | | | | |
|------|--------|------|-------------|--------------------------|-------------|-----|-------------|--------------------------------|
| 631E | 2A6B7C | LD | HL,(\$7C6B) | ; Routine to move Horace | A5C7 A5 | AND | L | ; Compare for keypress |
| 6321 | CDC26C | CALL | \$6CC2 | | A5C8 2002 | JR | NZ,\$A5CC | ; If not pressed jump over |
| 6324 | FE03 | CP | \$03 | | A5CA CBD1 | SET | 2,C | |
| 6326 | 2005 | JR | NZ,\$632D | | A5CC 2AAE81 | LD | HL,(\$81AE) | ; Get keyboard data for 'Dive' |
| 6328 | 3EFF | LD | A,\$FF | | A5CF 7C | LD | A,H | ; Load H with port address |
| 632A | 326F7C | LD | (\$7C6F),A | | A5D0 DBFE | IN | A,(\$FE) | ; Read keyboard |
| 632D | 3A6F7C | LD | A,(\$7C6F) | | A5D2 A5 | AND | L | ; Compare for keypress |
| 6330 | FEFF | CP | \$FF | | A5D3 2002 | JR | NZ,\$A5D7 | ; If not pressed jump over |
| 6332 | C8 | RET | Z | | A5D5 CBD9 | SET | 3,C | |
| 6333 | 32707C | LD | (\$7C70),A | | A5D7 2AB681 | LD | HL,(\$81B6) | ; Get keyboard data for 'Fire' |
| 6336 | C9 | RET | | | A5DA 7C | LD | A,H | ; Load H with port address |
| | | | | | A5DB DBFE | IN | A,(\$FE) | ; Read keyboard |
| | | | | | A5DD A5 | AND | L | ; Compare for keypress |
| | | | | | A5DE 2002 | JR | NZ,\$A5E2 | ; If not pressed jump over |

Starion

Starion uses quite an interesting method because you can redefine the keys. It uses a table to hold all the port addresses and the bit number in hex of the data line. So if you were checking for bit 3 then the instruction would be AND \$08.

Here is a snippet of Starion which reads the keyboard for some of the controls.

| | | | | |
|------|--------|-----|-------------|--------------------------------------|
| A5AC | 2AB481 | LD | HL,(\$81B4) | ; Get keyboard data for 'Bank Right' |
| A5AF | 7C | LD | A,H | ; Load H with port address |
| A5B0 | DBFE | IN | A,(\$FE) | ; Read keyboard |
| A5B2 | A5 | AND | L | ; Compare for keypress |
| A5B3 | 2001 | JR | NZ,\$A5B6 | ; If not pressed jump over |
| A5B5 | 0C | INC | C | |
| A5B6 | 2AB281 | LD | HL,(\$81B2) | ; Get keyboard data for 'Bank Left' |
| A5B9 | 7C | LD | A,H | ; Load H with port address |
| A5BA | DBFE | IN | A,(\$FE) | ; Read keyboard |
| A5BC | A5 | AND | L | ; Compare for keypress |
| A5BD | 2002 | JR | NZ,\$A5C1 | ; If not pressed jump over |
| A5BF | CBC9 | SET | 1,C | |
| A5C1 | 2AB081 | LD | HL,(\$81B0) | ; Get keyboard data for 'Climb' |
| A5C4 | 7C | LD | A,H | ; Load H with port address |
| A5C5 | DBFE | IN | A,(\$FE) | ; Read keyboard |

Here is part of the table which holds the keyboard data.

```
81B0 08 FE 04 DF 01 DF 08 7F
81B8 04 FB 00 00 00 00 04 FB
...
...
81A8 06 14A1 50 C3 50 02 FB
```

Even in a game as technically complicated as Starion the keyboard reading routine is still relatively simple and is quite compact. As you can see by filling in the table with you own values you can quite easily have the controls you want and of course this would have been done by the key definition routine. All in all, a neat and tidy method.

As you can see even the pro's don't make things more complicated than they need to be, but they do use some interesting tricks. I suggest using Spin to examine the code of some of your favourite games to give you some ideas or inspiration.

So there you have it, all you need to know about reading the keyboard. I have deliberately kept all the code plain, simple and easy to read with no coding tricks to make the code more slim line or faster. I leave that up to you.

All the diagrams in this article can be found on my web space
homepage.ntlworld.com/richard.chandler3

Enjoy, and if your sprites still look like blocks of melting ice bouncing round the screen at least you'll be able to control them!!!!

SPECIAL FEATURE

Poetry in motion: the making of Haiku 4K

By Matthew Westcott

"Old school" and "New school" are two terms that get bandied about a lot on the demo scene. Some people would have you believe that an old school demo is one that's done on an ageing platform such as the Speccy or C64, but it's much more complicated than that.

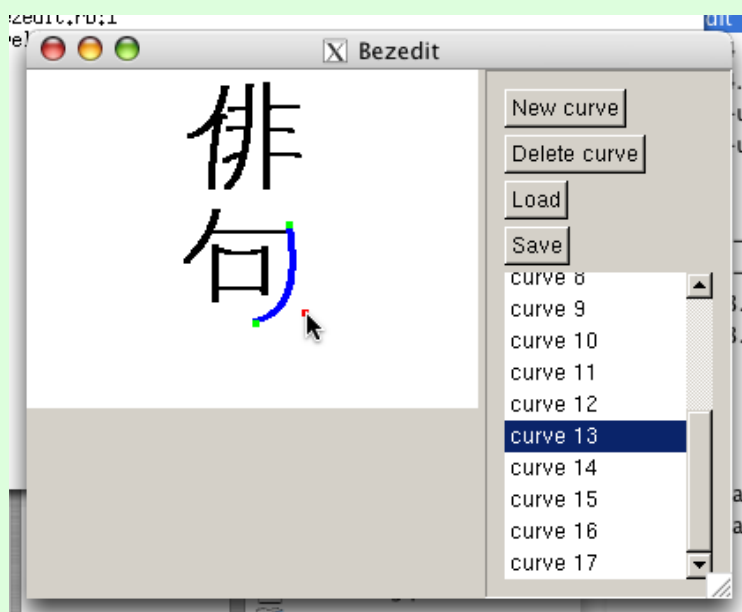
Roughly speaking, a new school demo is one that abandons the custom of all-out, in-your face, crowd-pleasing effects in favour of paying more attention to form, composition and design. As such, it's entirely possible to have a new school demo on the Spectrum, and I decided it was high time for another one. When German demo party The Ultimate Meeting announced an all-8-bits 4K intro competition amongst their roster of contests, I jumped at the chance - to the best of my knowledge, this was the first ever opportunity for the Speccy to take on its arch-rival platforms at size coding, outside of the Minigame competition.

The idea of a 4K intro based around Japanese culture had been floating in my head for some time. I've had a minor fascination with Japan ever since discovering jlist.com, an online shop selling an eclectic mix of Japanese items including some rather, er, interesting adult DVDs but also everything from Hello Kitty figurines to dried seafood to sprinkle on your sushi. As a nation they've perfected the art of making minimalism cool, which is just what you need when it comes to writing demos in 4K of code. But to be honest, my main reason for settling on Japan as a theme was that it provided a golden opportunity to rehash some old routines.

Character building



My previous intro, *How to recognise trees from quite far away*, was a rather optimistic attempt to implement an art gallery in 1K. The principle, as laid down by countless PC intros, was not to store the



actual images (which would have taken up 6K each) but the steps used to create them - in this case, a series of curves which can each be described by just three points, or six bytes. Bezier curves, to give the proper mathematical name. I went to the trouble of writing a mini art package called *bezedit*, and churned out half a dozen crude marker pen drawings, only to discover that 1K wasn't really enough to do the concept justice - the pictures still took up too much space and didn't compress well, and as a result I ran out of bytes much sooner than I'd hoped. It scraped third place out of four at the Forever Hex party. Still, it wasn't all bad. After a 15 minute tweak to the art package and Spectrum-side routines to make the brush strokes finer, they were successfully repurposed for drawing Japanese characters.

I made the decision to use genuine characters that meant something, rather than random squiggles, to keep it authentic and serve as a kind of 'Easter egg' in the unlikely event that any native speakers were to watch the demo. Reading up on the subject on Wikipedia, I learned about the three writing systems, *kanji* (to express things and concepts), *hiragana* (for Japanese names and grammatical constructions that don't have kanji) and *katakana* (for words borrowed from foreign languages). I soon found myself trawling through lists of kanji characters for hours on end trying to find a name for the intro - not only did it have to be catchy and pronounceable, but it also had to look good, and be relatively easy to draw.

I wanted to come up with something clever that combined ideas of Zen gardens and traditionalism with modernity, electronics and mathematics, but eventually I decided not to butcher the language any further and settled for the better-known concept of haiku - 17-syllable poems about nature. The idea of condensing a description of beauty into an ultra-

compact form was a perfect fit for what 4K intro coders do all the time - I could have kicked myself for not thinking of it sooner, really.

The credit line at the end, by contrast, was a much easier thing to get right - I was able to follow a few simple mechanical rules to translate the word 'gasman' into the katakana for ga-su-ma-n.

Music (and hats)



It's harder than you'd think to write tinkly oriental music. My first few attempts on my trusty electric keyboard (set to Koto or Shamisen mode) sounded fairly authentic, but stripped back to the square waves of the AY sound chip; they quickly turned into bubblegum dance music. Floundering in writer's block, I put on a DVD of Hayao Miyazaki's *My Neighbour Totoro*, a film so cutesy it makes Bubble Bobble look like *The Exorcist*. Lo and behold, after King Totoro had led the dance to make Mei and Satsuki's acorns grow, a melody leapt out, unmistakeably eastern and yet instantly appealing to western tastes. I analysed it every which way, pulled apart the plodding rhythm and the musical scale, and put together a melody of my own following the same style.

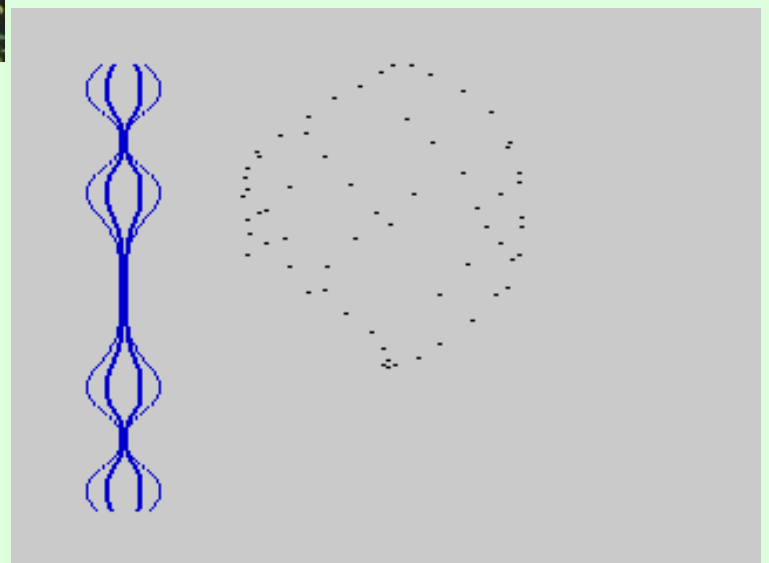
Traditionally, 4K intros on the Speccy have neglected the musical side slightly - even the all time classic 4Ks *Goa* and *C2H5OH* only went as far as a phasing bass line and a drum beat.

This is understandable really - coders can't call upon their local friendly musician for a soundtrack like they would with a full demo, because the standard tracker tools would take up too much space. (The *Soundtracker* player routine weighs in at 1086 bytes, and that's before you even start adding the actual music.) My approach was to effectively work backwards - coding a player to fit the tune I wanted to play - and that meant constantly juggling between

my musician hat and my coder hat. With the musician hat on, I'd take the music so far in a particular direction, and then as a coder I'd look for regular patterns and repetition to exploit in order to fit it into as few bytes as possible. As an example, all the instruments used in the track, including the drums, start out loud and fade away at a constant rate, so I was able to define each instrument in just two bytes - the initial volume, and the speed of decay.

Another space-saving ruse came out of confining myself to the oriental musical scale - I only used 20 different notes in the entire composition, and that meant I didn't have to include a big data table listing all the notes on the piano keyboard. Armed with that limited palette of notes, I was able to build up a 'library' of basic riffs that re-occur throughout the track - all the melodies, drum sequences and bass lines - totalling 100 or so bytes. By slotting these together in just the right combinations, I was able to create a gradually progressing melody, without storing any more copies of the main themes than absolutely necessary.

Making waves



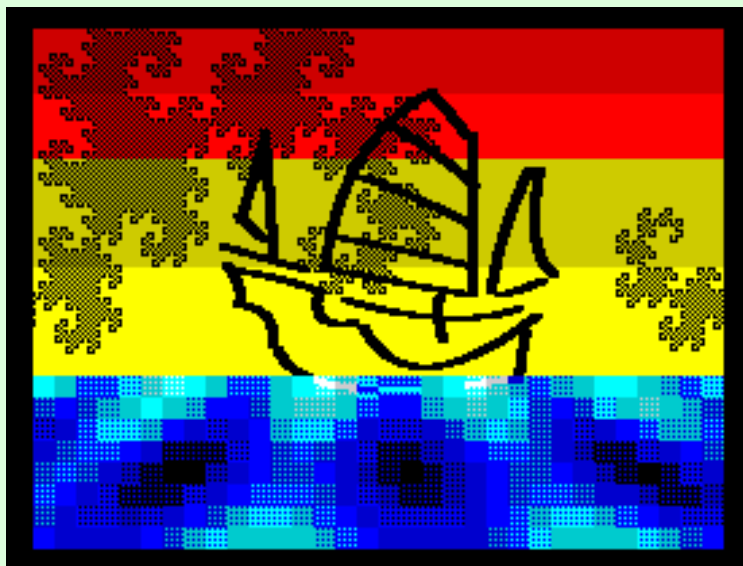
When the intro begins and the music starts, there's a short pause before anything appears on screen. This isn't just for dramatic effect. While everything looks calm on the outside, the Z80 is going fifty to the dozen pre-calculating some important things to be used later. To start with, it plots a sine wave, plundering the ROM routine to do the dirty work. Sine waves are useful things to have around - practically every effect in any demo ever is a creative abuse of sine waves. One thing we can do with them is make circles and ellipses, and if you stack lots of ellipses on top of each other you get a sphere. This is the second thing we do before the action starts—we create a 32x32 table that translates latitude / longitude co-ordinates into addresses of pixels on the screen.

BELOW FARTURE

Doing all this hard work in advance means that when we come to draw the sphere in part two, it's actually a fairly lazy job - look up a point in the table, plot it, jump along the table a bit and repeat 64 times.

As pretty as the sphere is, by itself it makes for a rather bare screen. It reminded me of the paper lanterns you might see at a Chinese New Year celebration (so not actually Japanese at all, then... oh well, that's artistic licence for you) so I thought about other decorations in the same vein - the ribbon effect fit the bill nicely. The code behind it is nothing to write home about - just a sine wave (natch) being vertically distorted by another sine wave - but it turns out to be a pleasantly effective little display.

Whatever floats your boat



When we're on the lookout for things that get maximum visual impact out of as little maths as possible, fractals are a good bet, and one type of fractal is particularly apt here - the dragon curve, which makes up the swirly cloud / tree things in the boat sequence. This little beauty is formed by starting off with an L shape, then replacing both branches of the L with two smaller L shapes pointing in different directions, and then repeating the process with smaller and smaller lines until you develop a squint or your pencil breaks. It's also what you get if you fold a piece of paper in half over and over in the same direction, then open up each fold at 90 degrees. As a practice run I wrote a Basic program to plot the curve in a cascade of recursive GOSUBs, which turned out so satisfyingly elegant that I felt compelled to post it immediately to comp.sys.sinclair.

```
10 PLOT 180,60: LET a=12: LET x=2: GO
SUB 20: STOP
20 IF a=0 THEN DRAW x,0: RETURN
30 LET a=a-1: GO SUB 20: GO SUB 40:
LET a=a+1: RETURN
40 IF a=0 THEN DRAW 0,x: RETURN
50 LET a=a-1: LET x=-x: GO SUB 20:
LET x=-x: GO SUB 40: LET a=a+1: RETURN
```



The subject came up at a CSS meet up in London that weekend, and in the following days Simon Owen was kind enough to have a stab at a machine code version.

This was a very welcome contribution indeed, as when I tried the conversion myself I introduced a bug which made it more of a drunken ant than a dragon. Initially I didn't have a clear idea of where said swirly thing would appear in the demo, but a scout around the art department of my local Borders store provided the answer. Browsing around for classical oriental imagery for inspiration, I found a book on Katsushika Hokusai, an artist whose woodcut prints will be familiar to anyone who's ever visited a Japanese restaurant. Fishing boats engulfed in fierce claw-like waves... Mount Fuji gracing the skyline in the distance... see, you do know him. The swirls of the dragon curve were a striking match for his detailed, foaming tsunami images, so I knew I simply had to try and recreate one of his ocean scenes.



Delving into the demo coder back catalogue of routines that could pass for water, I came up with a simple plasma effect (more sine waves added together...) with a bluey-white palette.

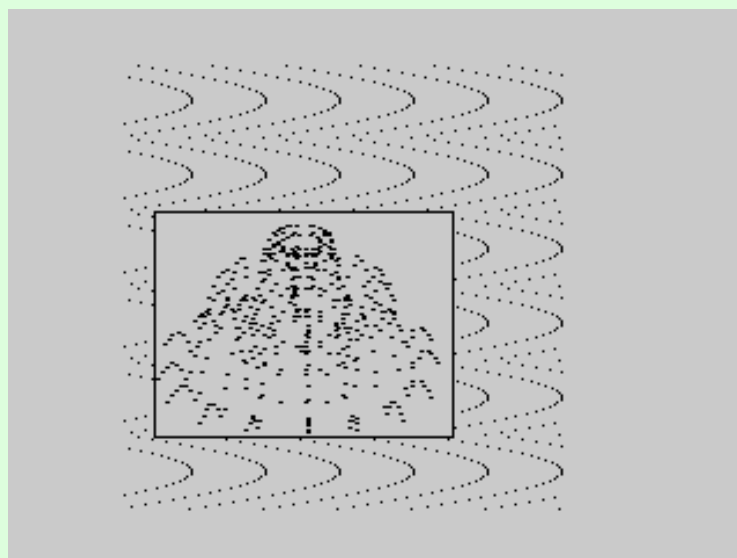
By increasing the wavelengths of the sine curves as I went down the screen, I was able to create a convincing perspective effect.

For the drawing of the boat, it was back to my hacked-up art package. The curve plotting routine was already there in the code, so it was only fair to milk it for all it was worth... All I had to add was the image data, which came in at 138 bytes, corresponding to 23 brush strokes. Not a bad deal at all. I was going to try and copy one of the original Hokusai prints, but for one reason or another my Rolf Harris skills weren't up to it. Instead, I found a suitably photogenic picture of a Chinese junk in Hong Kong on Google Images. (There goes the Chinese / Japanese artistic licence again.)

You can call me Bob



As I mentioned earlier, I originally planned to have some sort of "Zen garden" theme to the intro. In the end, the only place that idea really survived into the final product was the fountains in this next part. There's a cunning trick at work here, but one that's fairly well known in the demo coder repertoire. This

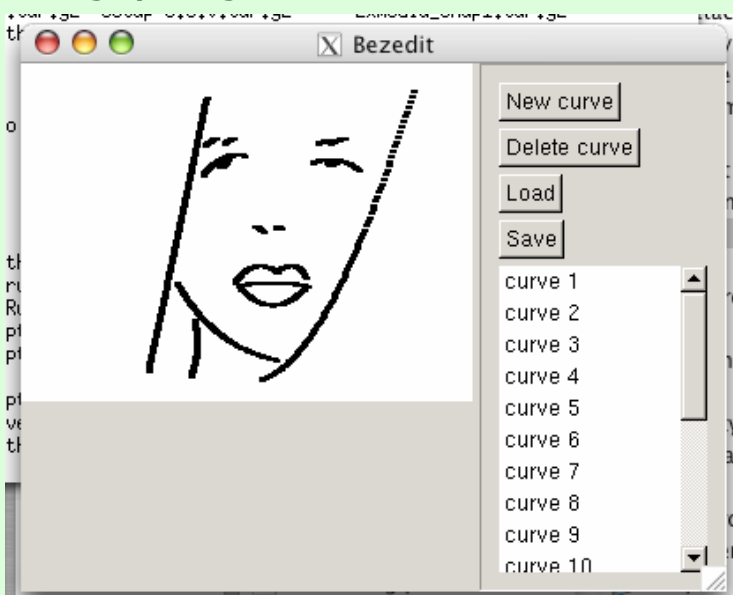


is the "Bob Demo", used whenever you have a string of things following the same path.

The most famous example on the Speccy is in Shock Megademo, where an endless supply of balls weaves its way around the screen - it looks as if they'd need to redraw hundreds of sprites on every frame, but in reality it's just a cycled animation of 8 or so frames. The only thing that needs to be added is the front most ball - everything else is just a repeat of that ball's previous position.

The same thing happens here, except the bit added to each frame is a ring of droplets instead. And that's all there is to it, really.

Giving up the ghost



With all this in place, I'd achieved pretty much everything I set out to do, and got it to fit neatly within the play time of the music as well. But did I manage to beat the 4K limit?

Well, yes, actually. Quite convincingly in fact. I still had a whole 1K to spare. There's nothing in the competition rules to say that you absolutely have to use up the whole space, but obviously the more you can cram in, the more chance you have of winning. But what could I add? I didn't have any immediate ideas for extra scenes, and anyhow, that would mean extending the music and juggling all my carefully crafted data lines around.

Besides, it was three in the morning, eight hours to the deadline, and I'd been awake for 23 hours. Normally, standard demo scene procedure in these circumstances would be to fill the remaining space with a long rambling scrolly message greeting everyone you could think of, but in this case it would have destroyed the ambience - the design of the demo dictated that all text had to be in Japanese,

with the English translations almost an afterthought.

This called for plan B - A Whopping Great Big Picture Of A Geisha.



I didn't have to look far for an image to copy, because this was around the time when *Memoirs of A Geisha* was in the cinemas, and posters for it were plastered on every street corner - I decided that would fit the bill nicely.

Unfortunately, back in my custom art package, my attempt at recreating that poster in Bezier curves turned out... rather dodgy. For this reason, I came up with the idea of having it fade in very briefly as a ghostly figure, just long enough to provoke the reaction "Ooh, it's a drawing of a woman's face..." without it turning into "...but a very badly drawn one".

It was also an excuse for a bit of nonchalant bravado, as if to say "oh yeah, I did this picture as well, but that's not important".

24 hour party people

ガスマン
g a s m a n

With the finished product packed up and ready to go, I finally settled down for some sleep and let the compo organisers take over. For the remainder of the party, I was able to return to the more leisurely pursuits of meeting people and drinking exceptionally cheap beer.

Mid-afternoon I was informed by one of the organisers that my 4K intro was the only entry in the competition, so we agreed to combine it with the 8-bit demo compo.

Some time later I was called over to a back room where we made a video recording of the demo to be shown on the big screen, and I proceeded to bemuse the onlookers by loading it onto the Spectrum from a TZX file on my Linux-powered iPod. (Trust me; it makes sense when you're packing a Speccy setup, a change of clothes and a sleeping bag into one rucksack.)

In the end I was up against two fairly run-of-the-mill demos for the Gameboy and C64, and won by a comfortable margin; at the prize ceremony I made sure to drop in a tongue-in-cheek remark slagging off the C64 scene for not putting up more of a fight. In keeping with the pirate theme of the party, my prize included a miniature treasure chest filled with chocolate money.

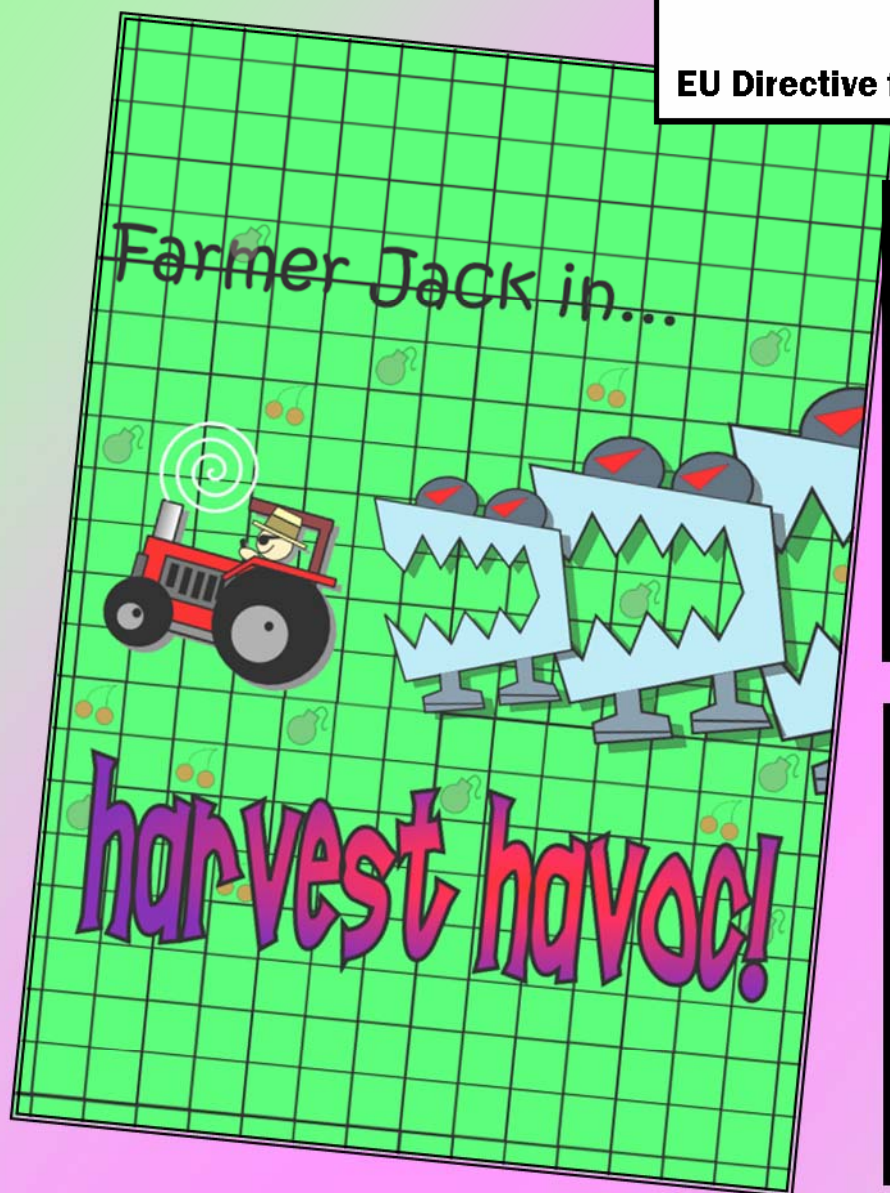
And so, the ultimate reward for my work was to be held up twice at airport security while they investigated what this shady figure was doing with a bag full of odd electrical equipment and "treasure". Ah, the joys of being a globetrotting demoscener... I wouldn't give it up for the world.

To download Haiku, as well as additional material including complete source code, visit:
<http://matt.west.co.tt/demoscene/haiku/>



**Driving a tractor whilst
drunk on scrumpy
can seriously damage
your health.**

EU Directive for pissed up farmers 10/06



Available on cassette for the ZX Spectrum

PRICE £2.99 (plus 99p postage)

COMING SOON FOR ZX MICRODRIVE!



www.cronosoft.co.uk

c/o RETROGAMES
Freiston Enterprise Park, Priory Road
Freiston, Boston, Lincolnshire PE22 0JZ
UNITED KINGDOM

LET'S PLAY

By Michael Bruhn

From this issue of ZX Shed and on I'll write a few pages about the Spectrum Zone of Home Action Replay Page (HARP). But before I go on telling you all what HARP is, let me introduce myself. My name is Michael Bruhn, I'm from Denmark, and was born in the Summer of '69. I've been a Spectrum user and mainly player since 1984. My biggest contribution to the Spectrum Scene was when I in the period of 1998-2001 was Editor of my own Spectrum Fanzine called Desert Island Disks. I've promised Lee to write an article about that Fanzine sometime in the future, so look out for that. I can tell now that I managed to get 7 issues out before I had to stop because of printing problems. Now I'm System Leader of the Spectrum Zone at HARP. This brings us back to what HARP actually is. It's a site where people compete with each other in computer and console games using various Emulators. As I write this it is possible to compete in games for the following systems: Astrocade, Atari 2600, Atari 7800, Commodore 64, Amstrad CPC, Gameboy, Gameboy Colour, Gameboy Advance, MSX 1, MSX 2, NES, SNES, Nintendo 64, Genesis and of course our beloved Spectrum. The site

works very much like Mame Action Replay Page, which I'm sure most of you have heard about. You make a recording for a game included in the HARP Spectrum Database, and submit it on HARP. Now it's ordered into its place on that game's Score chart, and you are awarded points for the Leader board for your effort.

Visit the Spectrum Zone for more information and rules:

<http://www.zxspectrum.homeactionreplay.org/index.php>

As I write this there are a total of 109 Regulation Play Spectrum games ready to compete in, and 14 Side Competitions. Regulation Play is the main competition in a game, while Side Competitions are used to score a game differently. Ex. Starquake, where you can play Regulation Play based on your final score, and a Side Competition based on how fast you can complete it. But as said, please visit the Spectrum Zone and take a look. If you want to register to HARP to start playing, go to the main HARP page:

<http://www.homeactionreplay.org/index.php>

Frankie - [Logout](#) - [Edit Profile](#) - [News](#) - [Board](#) - [Submit](#) - [Uploads](#) - [Logs](#) - [Search](#) - [Game List](#) - [Leaderboard](#) - [Stats](#) - [Info/Rules](#)
 ADMIN: [News](#) - [Country](#) - [Game Type](#) - [Company](#) - [Tournaments](#) - [Emulators](#) - [Games](#) - [Screenshots](#)
[Specy Tour 2005](#)

Home Action Replay Page - Search

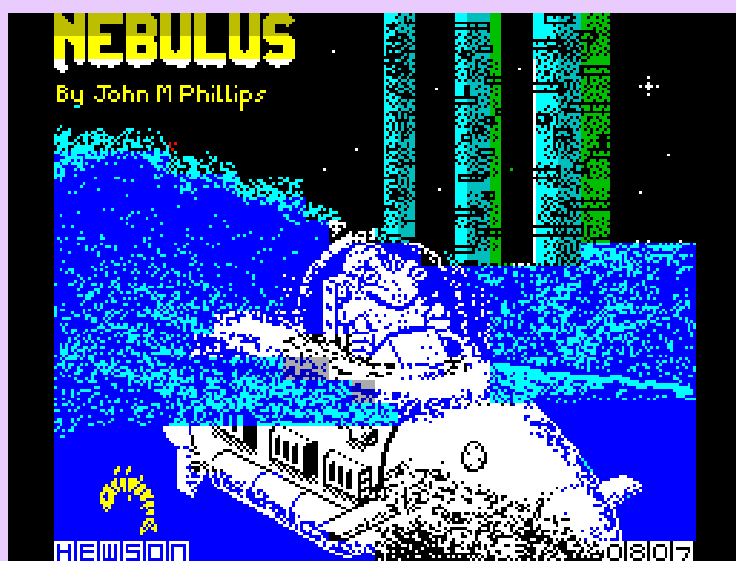
| Screenshot | Game / Tournament | Rank / Confirms | Username | Score / Stages / Length | Options | Uploaded... / Emulator |
|---|--|-----------------|---------------|--------------------------------|---|--|
|  | Bomb Jack Spectrum-Regulation Play Info - Scores | 1 C: 1 U: 0 | Frankie (DEN) | 1,462,070 35 00:22:22.00 | Download Utilities | 2006-01-23 13:23:02 SpectacWin Unknown |
| Specy Tour 2005 Entry. Position 4. Well, a good recording I think. No matter what I was the first to pass a million in the Tour on this game. Yippee :) [5.3] | | | | | | |
|  | Bomb Jack Spectrum-Regulation Play Info - Scores | 2 C: 1 U: 0 | DRN (ENG) | 943,190 35 00:19:56.00 | Download Utilities | 2006-01-23 11:52:44 SpectacWin Unknown |
| Specy Tour 2005 Entry | | | | | | |
|  | Bomb Jack Spectrum-Regulation Play Info - Scores | 3 C: 1 U: 0 | Tom (ENG) | 133,680 17 00:09:50.00 | Download Utilities | 2006-01-20 07:24:12 SpectacWin Unknown |

I didn't realize it was the last week of the tour. Posting my best scores so far, I know it's rubbish, but I haven't had much practice. Haven't seen this game before, likes it. (Time submitted is approximate, I couldn't be bothered to measure it precisely)

The Score chart for Bomb Jack.

So from this issue on, I'll bring you all an update on HARP's Spectrum Zone. It will consist of a review of one of the new games added to the Database since the last issue of ZX Shed. Then a follow up on what makes this game good for competition, or anything else of importance regarding this game on HARP. Finally you'll also get the most important news that has happened. I look forward to seeing you there.

Spectrum Zone Game Review:



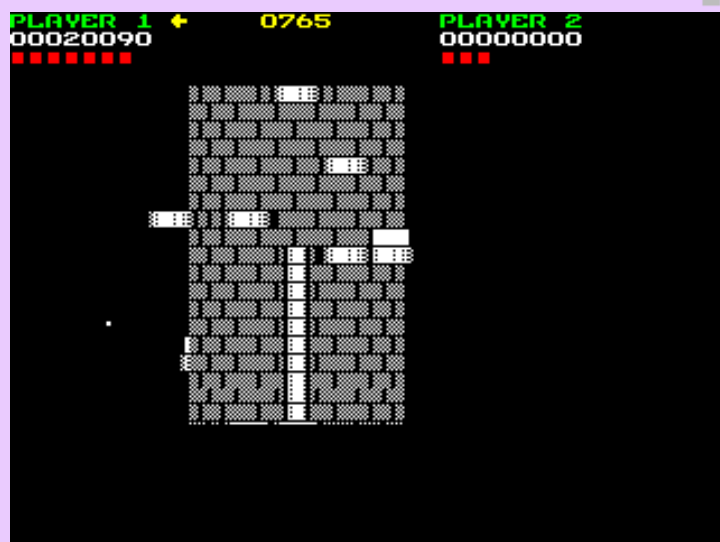
Nebulus

Hewson Consultants Ltd.

Original Price: £7.95

Release Year: 1987

You are Pogo, a hard working frog look-a-like character. You have been sent out on a job to destroy 8 huge towers on the planet Nebulus. To destroy the towers you must climb all the way to the top using the platforms, so that you can set off the destruction explosive. This is just your kind of job, but it seems this is not as easy as normal. The towers are guarded by all sorts of weird creatures who are trying to stop you reaching the top. On top of that it also not that easy to work your way up, as there are platforms that disappear when you step onto them, slippery platforms, a very confusing tunnel system, and elevators which to use you need to time right, so that they don't lead you into the arms of one of the creatures. There's also no time to rest, as you have to reach the top of the tower in a time limit, and one of the creatures are flying through the screen once in a while, and you have to avoid it by moving up, down or use a tunnel. Good thing is that you can shoot, as there are also some blocks that must be shot for them to disappear.



Watch out for those Eyes.

Then there are the jumping balls, which you can shoot (white), and some that you can only stun (black) for a little while. As said earlier you die when you don't get to the top in time, but you also die if you should fall into the water at the bottom of each tower. One little mistake, or being touched by one of the many creatures makes you fall down a few screens, and maybe even long enough down for you to disappear in the cold water.

Nebulus was John M. Phillips second game for the Spectrum after Hewson also released his first game Impossaball earlier 1987. They are both special in that they were totally new concepts for a game. All right, you could say that Impossaball just was a 3D version of Mirrorsoft's Action Reflex, but still. The graphics in Nebulus is great, and the smooth scrolling gives the game a great feel. Its one of those games, you never forget just because of the graphics and the atmosphere of the game. However the Sound is not much to cheer about I think.

Marks:

Playability – 8

Graphics – 10

Sound – 6

Instant Appeal – 9

Addictiveness – 9

Overall – 84%

Follow Up:

I think we can agree on that Nebulus is a great game, and therefore ideal for competing in. However the game has one problem. You get to many extra lives. For every 5000 points you get another life. Because of this you can easily leech a lot of points by climbing almost to the top of a tower, and then drop yourself into the water losing a life, and then

climb up again completing the tower. That way you collect the possible points for climbing a Tower twice, getting more than 5.000 points, meaning you get your lost life back, and get some free points. So therefore we had to come up with a scoring system that prevented this for Regulation Play, and came up with this if you complete the game:

Final Score + 100.000 - (X * 10.000)

X = Number of lives lost in the game.
100.000 bonus for completing the game.

Using this scoring system you lose 10.000 points for every lost life. Therefore it's not a good idea to do the climb top, drop down into water trick, as you lose more than you gain in the end. If you don't complete the game, it's just your final score that counts without the bonus. We have also set up a Side Competition where you can leech as much as you like. This is what Side Competitions are for on HARP. Under Regulation Play we try to find the best way to play the game to make it as interesting as possible, and following the spirit of the game.

I have to admit that back in the days I wasn't able to complete Nebulus. It was due to the competing with my friend Skito that I made tower by tower in a race to see who completed it first. He won in the end, and did also end up with a better score than me in both Regulation Play and Side Competition. But hey, I'll be back one day, I hope. Check out the momentary Score charts for Nebulus at:

Regulation Play: <http://www.homeactionreplay.org/search.php?gamename=Nebulus&tourney=2>

Side Competition (Leeching):
[http://www.homeactionreplay.org/search.php?gamename=Nebulus%20\(Leeching\)&tourney=13](http://www.homeactionreplay.org/search.php?gamename=Nebulus%20(Leeching)&tourney=13)

If you would like to compete in Nebulus on other systems, or just check them out, then it's possible to do that on HARP on the Amstrad CPC, and on the Atari 7800. On the Atari 7800 it was called Tower Toppler. Of course the Spectrum version is far superior to those versions :)

News :

In this section I'll bring you news from the Spectrum Zone since last issue of ZX Shed, but as this is my first article I'll just give you some recent news. The Speccy Tour 2005 was hosted on HARP, and ended in January this year. The winner was Aleksandar Lukic. He won the Tour for the third time, and ended up with 855 points, 146 more than second placed rebufo. Here are the top scores from the Tour:

| | |
|----------------------------|------------|
| Bomb Jack | 2.357.380 |
| Pyramid, The | 136.373 |
| Worse Things Happen At Sea | 105.730 |
| Splitting Images | 616.800 |
| Hyper Sports | 684.000 |
| Wec Le Mans | 577.060 |
| Splat | 53.710 |
| Bounder | 10.684.360 |
| Pinball Wizard | 50.690 |
| Glug Glug | 182 |

The Glug Glug score might not sound as much, but that's because we scored the game by number of levels completed instead of the score. Why this? Well, simply because it's too easy staying on the first levels leeching endlessly points, which would mean long and boring recordings.

The high score in Bounder was by our winner, and he played for more than 4½ hours, can you believe that? Well, if you want to see any of the recordings from the Tour, or some of the Top 3 recordings from previous Tours, go to the Speccy Tour section of HARP:

<http://www.zxspectrum.homeactionreplay.org/tour/tindex.php>

That's it for the very first Let's Play article. I hope some of you out there have ended up wanting to come over and compete. Maybe you'll end up in one of these articles, if you submit some great scores. See ya.

"This is uncanny. You drop in out of nowhere like Elite Viper ships and descend on Kizza like there's no tomorrow." - Palmer Eldritch

And the Troll and a Half (TAAH) award goes to...*drum roll*... Mr. Kizza for meeting the high exacting standards required to win this award including but not limited to shameless lying, pathetic attempts at deceit, bringing disrepute to the scene, name calling on other forums, twisting facts, et al.

It all started when Kizza made this innocuous statement in the Atari Age forums during a poll of *Best 8-bit gaming computers*. He said, "I had a few on cover tapes on magazines and a few published by budget producers."

He went on to claim he had written "Halls of Zatrex, Centipod, Zombie!, Advanced Lawn-mower Simulator, Steve Davis Snooker (renamed by publisher CDS originally called plain Snooker), Penalty Shootout, Forest, Big Boxing, Shooting Range and Digger Doug (which had digitised speech and music on the 128k!)."

He would have probably gotten away with that claim if it hadn't been for Elite Viper Squad (EVS) member Malc74 patrolling the forums for trolls. His quick background check on Kizza raised his suspicion when he found that he couldn't trace any of the Kizza games and the fact that Steve Davis Snooker was a well known title coded by Mike Lamb on the ZX Spectrum. From Kizza's date of birth in his profile, Malc74 also brilliantly deducted that Kizza would have had to be 7 years old at the time SDS was published!

Things went from bad to worse from thereon for Kizza as Malc74 called for backup at WoSF to set things right and pretty soon EVS members began uncovering more lies and false claims from Kizza including claiming authorship of a few Atari ST titles. EVS member Sard has already shown up Kizza on some of the ST titles he has claimed his own and we can expect more fireworks in the future as investigations go on.

For more information and detailed first hand accounts of proceedings please do check these threads:

The original Atari Age thread:

<http://www.atariage.com/forums/index.php?showtopic=91776&st=25>



The WoSF thread on Kizza

<http://www.worldofspectrum.org/forums/showthread.php?t=12790>

On a more serious note, many people spend a lot of their time archiving as much information as possible. For someone to be claiming authorship of titles makes this task more difficult. Fortunately, in this case, there was no question that he was lying. Had he been more convincing, we could very well have had false data in infoseek.

Archive – Website Update

There have been a number of updates to the archive site thanks to getting hold of a entire collection of *Popular Computing Weekly* magazines. Twenty six missing in action titles have been added from the small line adverts. If you placed an advert in PCW for Ace software you might have got one of our contact letters because we have written to every advert. Even after 23 years! We got a reply which we very excited about because we may have found someone with the JET-Disk system that was made for the Ace back in 1982.

A new section on the site has been started called customised Aces. A few JA owners over the years added modifications to there JA's we start with details of three Aces with another four to come later just as soon as I put the web pages together.

Other Forth documents have also been added that are about the same age as the JA's. Theses make great reading to see how Forth has changed over the years.

We also have our own Forums, so if you have anything to say, or want to know, then the Forum is the place to ask. If you have any material that you can help with then please make contact via the site.

www.jupiter-ace.co.uk

The 99K Jupiter Ace

What? I hear you ask 'a 99K JA'. Most people know of the JA as having 3k on the machine, almost 1k is taken up with the screen display, another with system variables and other chunks of bytes used by the Ace for character ram leaving the user with about 889 bytes to program with. It does not sound like a lot but Forth is more compact than BASIC. Adding more RAM is the only option to define larger Forth dictionaries and this is how we get the odd RAM values of 19K and 51K. The built in 3K in the Aces system is below address 16384 and if you add a 16K Ram pack you get the total of 19K or 51K should you be lucky enough to find a 48 JA RAM pack. The EB 96K RAM pack is a modern solution a two chip interface. The RAM pack is based around the RAM chip 628128 128x8 bit and a 74LS153.

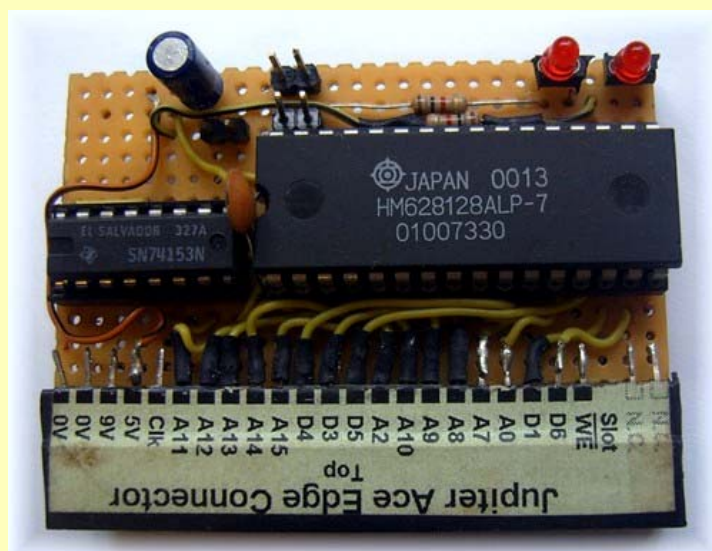
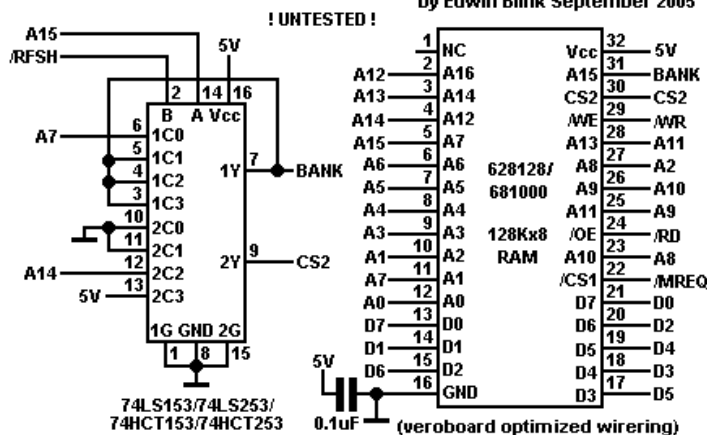
How it works; The LS153 is a dual selector chip. We use one half as an OR gate to do the address decoder the other half as feedback output to create a latch. This is loaded during a Z80 refresh, A7 is bit 7 of the Refresh register. This can be programmed by software using LD A,128 LD R,A which selects the other page of 48K of RAM.

The 128K ram is accessed from 16384 to 65535, your bankswitch allows two pages at that range and

internal ram is not paged out. To use the extra 48k page the stack needs to be relocated.

Jupiter Ace 96K RAM expansion

by Edwin Blink September 2005



The 96K prototype

As you can see in the above image Led's have now been added to show which RAM page is currently in the memory locations 16348-65535.

While testing we found a small problem with the RAM paging hardware. When paging the voltage on the bank signal was below 3.5v. This did not give the logic 1 that was needed for the second page to work. This has been fixed now, by Edwin. A 1n (1000pf) capacitor needs adding between the BANK and 0v line.

To use the RAMpack as a 48K extension you must do the following after you power up the Ace:

65535 15384 ! QUIT

This tells the Ace where to store the return stack. You can use the RAM now as any normal 48K RAM expansion with the Aces Forth.

96K Memory Paging words in AceForth

First clear the RAMTOP to below 16384:

```
16383 15384 !
```

Then type in the following three words of Forth.

```
DEFINER CODE DOES> CALL ;
```

```
CODE PAGE0 175 C, 237 C, 79 C, 253 C, 233 C,
```

```
CODE PAGE1 62 C, 128 C, 237 C, 79 C, 253 C, 233 C,
```

Now you will have two new 'AceForth' words PAGE0 and PAGE1. Which will swap the two 48K pages, the LED will be bright to show which page is currently in use.

NOTE: That when you swap the pages you also swap the return stack. That's why we lowered the RAMTOP to below 16384. You could make a Forth word to move the return stack to the PAD, then swap pages, then copy the return stack to the new memory page. One other thing! If you have an AceForth word definition which crosses over the 16384 page boundary you will have a crash when swapping pages. A possible solution is the reserve a few bytes of RAM which crosses the page boundaries. We will be looking at these words soon

With the EB RAMpack now debugged and working correctly we have add it to the ACF. The ACF is the Jupiter Aces Compact Flash mass storage device.

The ACF is now in a version 5. Edwin Blink has just written the Formatting code. It's was great to see the Ace formatting 64Meg Kodak CF card. More details next, time about the MFS;

Edwin's Micro File System for the Jupiter Ace and how we are going to include the new saving and loading of Forth programs with the ACF.

ACE Forth

Forth is a programming language that uses two stacks and a dictionary of words that a programmer adds to in writing a program. ZX BASIC users write their programs using line numbers, the BASIC is then added to each line as the ZX BASIC program grows more lines of BASIC. The program is then usually run from the first line. Jupiter Ace programs are stored in the memory as Word's in a DICTIONARY.

WORDS are what make up Ace FORTH, a collection of FORTH WORDS can be saved to tape as a DICTIONARY - We do not use the term program.



I am sure we have all done something like this ZX BASIC 10 PRINT "Steve "; RUN 10 and Steve gets *Jupiter Ace ACF version 4*

printed to the screen. In Ace FORTH its

```
: greet ." Steve " ;
```

The ':' is how you start to define a word in all Forth's and the ';' ends a FORTH word definition. The 'greet' is the name of the FORTH word and this is used to call the commands contained with in the ':' --- ';' Next in our greet word is the FORTH word "." (Dot quote) note that there is a space after the dot quote. This is in all Forth's and prints the following characters to the screen up to the end marker, which is the closing quote. Next is a space, then the end of a word marker ';'. To call the word we enter the name greet, as in all Forth's the Jupiter Ace responds by echoing the name greet to the screen. Next, greet carries out its FORTH words printing 'Steve' to the screen. The Ace then prints OK to indicate that its finished task.

Once defined, greet can be used in other FORTH words as its now part of Ace FORTH. So, we could do;

```
: big_greet greet greet greet ;
```

Here a new word has been defined called big_greet it uses the FORTH word greet, which was defined before it in the Jupiter Aces dictionary. Big_greet will display the message three times because greet is called three time.

Next time; The Stacks and numbers. If you would like to mess about with AceForth then download the brilliant emulator Eighty One from WOS.

A Math Co-processor for the Jupiter Ace

Professor J.V. Noble
Institute of Nuclear and Particle Physics
University of Virginia
Charlottesville, VA 22901 USA

This note reminisces about the author's first digital design project. Despite his initial unfamiliarity with the world of digital electronics, the project took but two weeks from start to finish, working part-time. Much of this celerity can be attributed to the fact that the host computer was programmed in ROM-based FORTH.

In 1980 and 1981 I acquired my first personal digital computers, the Sinclair ZX81 and the Timex-Sinclair 1000 (same machine, different name).

Although vastly slower at floating point arithmetic than the CDC mainframe at work, they were nevertheless useful for serious calculations because of their rapid turnaround: they were interactive, and their time belonged solely to ME.

In fact, a paper I published in 1982 noted that "...all calculations in this paper were performed on a Sinclair ZX81 computer"--one in the eye for a competitor who had gotten the wrong result on a huge IBM mainframe.

The Sinclairs were remarkably simple Z80A machines that ran at 4 MHz, stored programs on an external cassette tape recorder, and used a standard TV as a 25 line by 40-character display. The membrane keyboard was, bluntly, horrible; the display hard on the eyes; and the entire lash-up fragile beyond belief--the slightest jar produced power or data glitches that instantly vaporized an afternoon's work.

About the same time I received a flurry of consulting commissions from local attorneys, dealing with accident investigation and reconstruction. Whereas the Sinclairs were adequate for small programs that ran a long time, the programs needed for accident reconstruction were more complex and more computation-intensive. Typical cases ran an hour or more. It seemed to me that both for use at home, and (eventually) in the field, a faster, more stable machine was needed. Manifestly one could easily make a Sinclair (trans) portable: a car TV and a 12 Volt cassette power supply could plug into a cigarette lighter, from which the computer also could be powered. However I doubted such a rig would work reliably amidst the bumps and grinds of the automotive environment; and anyway it was not fast enough by an order of magnitude to explore the necessary range of parameters during in-the-field

reconstructions of accidents.

As I sat pondering my next move, advertisements began to appear for a new machine, the Jupiter ACE ('Automatic Computing Engine') by Jupiter Cantab, Ltd. It was touted as 10* faster than the ZX81 because it employed a compiled language, FORTH, rather than an interpreted BASIC. The machine was distributed in the United States by Computer Distribution Associates (CDA, henceforth) of Pittsford, New York. (For those unfamiliar with the more arcane geography of New York State I should add that Pittsford is a small neighbour of Rochester, that Babylon of iniquity lying between the Genesee River and Erie Canal.)

My ACE cost a deuce--\$200--but I was happy: it was indeed 10* faster than the Sinclair, especially at floating point arithmetic, despite being essentially the same 4 MHz Z80A machine. So what if the internal representation of floating point numbers was a strangely bastardized BCD? So what if floating point and integer data shared a common stack? And so what if the developers had neglected to provide trigonometric, exponential and logarithmic functions... Uh, oh: programs that describe colliding cars need those functions. How could I get them?

Settling down resignedly to a tedious session of programming table look-ups I was struck by a happy thought: perhaps the jolly old chaps at Cambridge, England intended a math co-processor add-on to plug into the expansion port on the backside of the ACE. I called CDA to inquire if such might be the case, pointing out that many like me might purchase an FPU accessory if available. The engineer at CDA said Jupiter Cantab had no plans for such an item, as floating point math chips were unavailable abroad. He then inquired whether I would be interested in designing one and licensing it to them. Flattered but flabbergasted best describes my state of mind. I temporized, requesting a week to consider whether the commission lay within my competence. During that week I was able to acquire spec sheets for Intel 8231A and AMD 9511 (equivalent 8-bit math chips).

I also discovered that supply houses wanted serious bread for these chips in quantity 1. The rest of the week I spent reading the relevant sections of Horowitz and Hill, *The Art of Electronics* (Cambridge University Press, Cambridge, 1980). This remarkable book convinced me that it, coupled with my extensive practical experience of analogue electronics, would see me through. This is how I came to design the math board that is the subject of this reminiscence. My only stipulation was that CDA should supply the rather expensive math chip.

The hardware presented several design problems. First, the HMOS chips demanded a dual-voltage (5-

and 12 V) power supply. A regulated 5 V line was available at the expansion bus. Where to find +12 V? Eventually I settled on voltage-doubling the unregulated +9 V line of the expansion bus using the circuit of Fig. 1.

The next step was to provide data transmission and chip control. The math chip's 8 data lines were interfaced via a 74LS245 bidirectional tri-state bus driver, to avoid excessive loads on either the CPU or math chip. This driver requires two signals, CE, and S/R. When CE is low data can be transmitted, S/R controlling the direction. When CE is high, however, the two sides are disconnected.

The FPU was controlled via the Ace's IORQ, RD, WR and WAIT lines. The latter is pulled low by the FPU to indicate it is busy, and is used to prevent the CPU from attempting to read or write. (I eventually found it necessary to provide an open-collector buffer for this line using two transistors, to prevent the CPU from being locked up by the FPU's WAIT line.) Finally, I used 3 port addresses, 251, 253 and 255 to provide the signals for reading and writing data, and for resetting the FPU. As I had never designed a digital circuit before; the address decoding and control circuitry was unnecessarily complex (Fig. 2). If I were designing the circuit today it would be at least one 'glue' chip simpler.

Nevertheless, IT WORKED! The breadboard version was completed in two weeks of part-time labour. I made heavy use of FORTH's ability to address directly the data ports, in testing the address decoding and control signal logic. That is, the breadboard was assembled incrementally to avoid dangerous inputs to the expensive FPU. Once the math chip was added, its operation could be tested immediately using 16- and 32-bit integer arithmetic (of which it was capable). Simple unary floating point operations such as square roots of perfect squares, and logarithms of powers of 10 further confirmed that the FPU was operational. (The designers of the 8231A thoughtfully provided direct conversion of integers to IEEE floating point and vice-versa.) During the testing period I used only high-level FORTH--involving IN and OUT--as I was by no means confident of my ability to write CODE definitions.

True floating point software presented difficulties. Jupiter FORTH contained unfortunate choices such as mingling floating point numbers and integers on the same stack as well as a non-standard IEEE-Incompatible representation for floating point numbers. Worse, I was at that time a FORTH tyro who had not yet learned to use the power of the language. It took an additional month to write the revised floating point lexicon. Most of the work involved translating their fp representation to IEEE and vice-versa.

Every story deserves an epilogue, or in this case, an

epitaph. When I reported back to Computer Distribution Associates with a completed design, I discovered to my dismay that they had in the meantime performed their famous dead-guppy imitation: belly up. So had, Jupiter Cantab, Ltd. No more ACEs, no royalties, in fact nothing but the Circuit board shown in Fig. *3a, and my (somewhat modified) ACE shown in Fig. *3b. Finis.

**SPT- images 3a and 3b have been lost over the time*

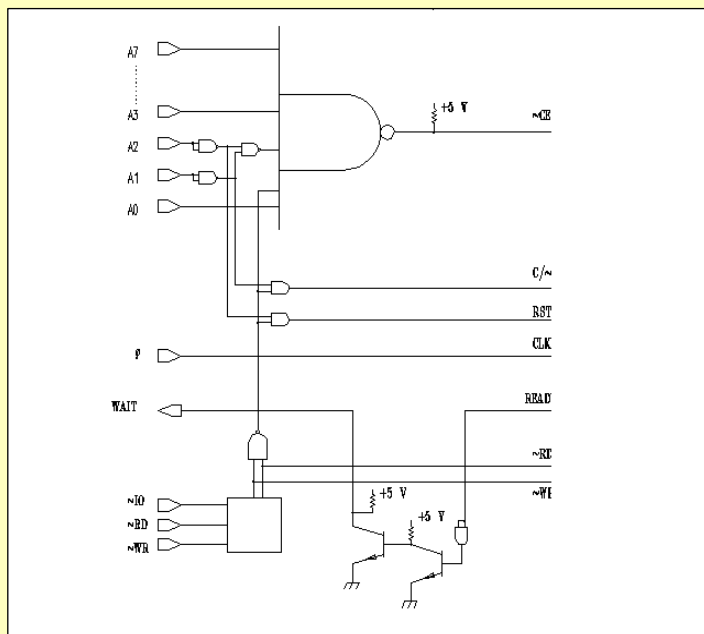


Fig 1

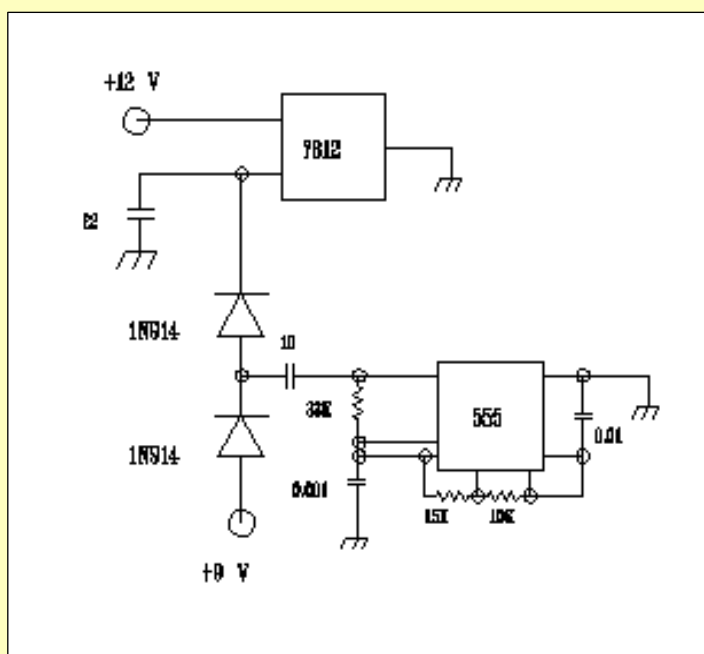


Fig 2

Many thanks to Professor J.V. Noble from the Institute of Nuclear and Particle Physics for sending this item to the Jupiter Ace Archive Project – SPT/EB

Well, that's it until next time...

May the FORTH be with you!

**LEGENDS MAY SLEEP
BUT THEY NEVER DIE!**

**OFTEN COPIED
→ FOREVER INCITATED →
NEVER EQUALLED**



ACCEPT NO SUBSTITUTE

[HTTP://WWW.WorldOfSpectrum.org](http://www.WorldOfSpectrum.org)