

DEVASTATING (FINALLY HERE) NEW LOOK ISSUE!

Free PDF Download
Check Website For Hardcopy
Details



ZX SHED

reloaded

MK-14
ZX-80
ZX-81
ZX SPECTRUM 16K
ZX SPECTRUM 48K
ZX SPECTRUM 128K
QL
ZX SPECTRUM +2
ZX SPECTRUM +2A
ZX SPECTRUM +3
Z88
JUPITER ACE
SAM COUPE
PC 200

WE ARE BEHOLD

*And this time...
We're Bi-Monthly!*



**THE WOS'ISH
SINCLAIR COMPUTERS
FANZINE FINALLY RETURNS**



hiive

BOOKS

RETROGAMING BOOKS FOR THE DISCERNING READER

CONFIRMED FUTURE TITLES!

Generation ZX (2009)
Generation 64 (2010)
The Amiga Book (2010)
The SNES Book (2010)

IN THE PIPELINE...

The Game Boy Book (201x)
Generation 8-Bit (201x)
The Arcade Book (201x)
8-Bit Worlds (201x)

<YOUR BOOK HERE?>

AVAILABLE NOW!

The Commodore 64 Book - 1982 to 199x
The 8-Bit Book - 1981 to 199x (softback)
The 8-Bit Book - 1981 to 199x (limited-edition hardback)

www.hiivebooks.com
(follow us @ twitter.com/hiivebooks)





ISSUE 1 - JULY 2009

EDITOR AND CRAYONS

Melissa Ward

EXECUTIVE EDITOR

Lee Fogarty

CORE WRITING TEAM

Richard Chandler

Skarpo

Ian Hamilton

Steve Parry-Thomas

Arjun Nair

Ben Rapier

ADDITIONAL CONTRIBUTORS

Paul Johns
Andrew Ryals

CONTENTS

48/128

05 – EDITORIAL

Some random ponderings from the new editor.

06 – NEWS

Keeping up with the latest Spectrum Goodness.

08 – REVIEWS

We take a look at Egghead IV, Farmer Jack, Loco Bingo and a rather nice book to name a few.

15 – CODE SHED

Part IV of Arjun's guide to BASIC

20 – LOST TREASURES

Ian Hamilton takes a look at 'The Archaeologist and Babaliba.

21 – PROGRAMMER'S DIARY

More from Jonathon Cauldwells journey creating **LOCO BINGO**

25 – GOING COMMANDO

Part two of the complete guide to hacking the elite classic.

30 – INTERVIEW

Ben Rapier speaks with Richard Jordan of **RETROSPEC**

32 – SPECIAL FEATURE

Carl Looks at the awsomeess of **THIEVES SCHOOL**

38 – JUPITER ACE

Steve(spt)'s further adventures in the world of the Jupiter Ace

41 – PREVIEWS

Keeping an eye on the latest Spectrum Goodness on the horizon



ADDITIONAL MATERIAL BY KIND
PERMISSION
WWW.WORLDOFSPECTRUM
WWW.THE-TPSHOP.COM



**EL HEROE DEL SAIMAZOON EN UNA
AVENTURA TOTALMENTE NUEVA: BABALIBA**



15.000 pts. al primer lector que nos envíe el secreto de **BABALIBA**

Distribuidor
exclusivo para
España:
MicroWorld,
tlf: 441 12 11
(Dto. a tiendas: 40%)



Pedidos
contra reembolso
e información a:
«Mansión DINAMIC»
C/ TILOS 2, N° 21,
Montepríncipe,
Boadilla del Monte,
MADRID
(sin gastos de envío)

EDITORIAL

WELCOME

48/128

Go on, admit it, you're glad to see us back ain't ya? And probably a little surprised too, after all, it has been a while... more than a while... okay it's been bloody ages but the fact remains we're back and this time we're here to stay!

You can raise your eyebrows as much as you like, believe it, receive it, the Shed is back, bigger, bolder, better than ever and as you may have noticed from the front cover - we're now going bi-monthly.

Woah there, yeah, well, big promises from the fanzine that managed an issue a leap year if you were lucky, the fanzine that's concept of time would even baffle Stephen Hawkin, well, you just wait and see. From this issue on as long as there is a *World of Spectrum*, as long as there are Sinclair enthusiasts out there and as long as people actually are interested in what we have to say there will be a Shed. A ZX Shed.

Exciting huh, well, a lots happened around here since the last issue so we had better get going.

Bit of a sad start to the celebrations I'm afraid, **Lee Fogarty**, Master and Commander of the good ship Shed has had to stand down from his duties due to having a real life. Something most of us aspire to have one day and we all wish him a speedy recovery. The good news is he's not gone and will be keeping a watchful eye on us to make sure we're all behaving ourselves.

So, on with the new, what can you expect from the new look Shed? Well, for starter we can assure you all those fantastic articles we began long ago will be finished. Oh yes, we will continue and conclude **Arjun's guide to BASIC**, and the complete guide to hacking the elite classic Commando in **Going Commando from Mr Anonymous**. We also have a wealth of material that has been supplied over the years that has been waiting in the trunk, I can assure you it will all soon see the light of day, nothing will be rejected or wasted. Plus we will have all the usual news, reviews, interviews and articles you expect from

us... Plus a few new features along the way that we're all rather excited about.

I would also like to point out the original run of **ZX Shed** is still available in PDF download from the *World of Spectrum* archive, and that if you go to the forum there you will find a dedicated **ZX Shed** area for you to leave us messages and thoughts you would care to share.

Also, I would like to draw your attention to the fact that we also now have a shop! Oh yes, you can now buy to wear and display with pride lovely Shed and Spectrum related tat at rather excellent prices. Oh yes, T-Shirts, Mugs, Mouse mats... it's all here. Go on, indulge yourself at

www.cafepress.co.uk/zxshed

Well, that's about it from me. Hope you like the new look and we all look forward to seeing you in the forums.

Later Peeps

Melissa Ward
Editor

NEWS

48/128



'CRASH' and 'Your Sinclair' Legend Graeme Kidd dies aged 52.

As we were putting the final touches to this issue we learnt of sad passing of Mr Graeme Kidd, Former mayor of Ludlow, driving force behind the town's internationally renowned food and drink festival but better known to us for the time he spent on the classic Sinclair ZX Spectrum publications 'Crash' and 'Your Sinclair'

Mr Kidd suffered a suspected heart attack on the morning of Sunday 31st May and sadly passed away despite efforts to save him.

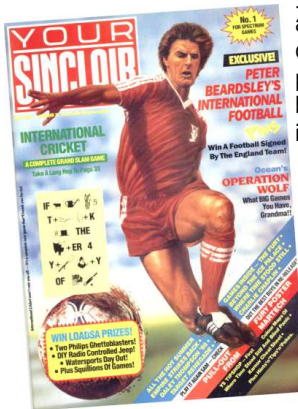
He was a prominent public figure in Ludlow and the

What a Silly Hunt – August 1985 Graeme Kidd takes over as editor of CRASH, and remains there for the next 19 issues.

brains behind many of their public events including the new Spring event that attracted business to the area.

His funeral was held on Tuesday 16th June at St Laurence Church in Ludlow.

He is survived by his wife and two sons.



Source – NickH (World of Spectrum Forums) + shropshirestar.com

ZX SHED RETURNS

Well, as you can see, we're back and all that, but this time things will be a little different, what's that? Different? Well, apart from the fact we aim to deliver issues on time we also want your support, oh yes, this time out it's all about you guys, we want your articles, your thoughts, ponderings and musings. We are a community publication so please get on board. We've been labeled 'WOS'ish' now it's time to prove it.

SO, fancy writing an article? Got something you want to share? Well, let us know on the forum and let's get the ball rolling.



Retro Gamer 65



Page 99 people!

As you may, or may not know already Retro Gamer number Sixty five has a rather fine article with our very own Mr Fogarty about the original incarnation and run of the SHED! Mighty fine read, even though they did spill the beans regarding our return (and they did some freaky stretchy things with the pictures! ED.) Go check out a copy now!

The 8 bit Book – 1981 to 199X

The next publication from the buzzing hive of retro based books is almost upon us! **The 8 bit book** by Jerry Ellis will be with us early this month, and by the looks of the previews on the publishers site its shaping up to be another great read.

The book will take a nostalgic look at over two hundred titles spanning a wealth of machines including the **ZX81**, **Oric** and **Amstrad CPC**.

Needless to say you can expect a full review as soon as we receive our copy! In the meantime be sure to check out HIIVE Books website at <http://www.hiivebooks.com> for the latest updates.

The 8bit Book will cost £17.99 as is available for pre-order now.



Retrochallenge 2009

Retrochallenge 2009 begins on the 1st of July and will run for 1 month, from the website –

The origins of the RetroChallenge are as indistinct as its purpose.. some say it started in 2005; others swear that it goes back as far as 2004. Those who know for sure are unwilling - or unable - to say.

RETROCHALLENGE 2009

A BIG STEP BACKWARD FOR PERSONAL COMPUTING

About

Hey! Ho! Let's Go!

Rules

Yes indeed, fellow retrocomputing enthusiasts, koders and sadists alike, it's time for everyone's favourite semi-organized, non-combatd, retrocomputing showdown!

Enter

Updates

Links

Blog

That's right: July 2009 is time for what may well be the 6th annual RetroChallenge (though no one's quite sure about that). Time to dust off the '18-48 and finally finish that text adventure you started programming in grade 7, or by doing all your online shopping with a Commodore 64... whatever.

In addition to the usual nonsense, we've also got a HUNT tournament organized, courtesy of **Robbley**. So if you feel the urge to massacre your fellow challengers (and really, who doesn't?), now's your chance.

©

RetroChallenge is brought to you by:
68k Macintosh Liberation Army



In a nutshell, the Retro-Challenge is a loosely disorganised gathering of RetroComputing enthusiasts who collectively do stuff with old computers for a month.

The event is very much open to interpretation... individuals set their own challenges, which can range from programming to multimedia work; hardware restoration to exploring legacy networking... or just plain dicking around. It really doesn't matter what you do, just so long as you do it.

While the RetroChallenge has its competitive side, it's not really a contest... it's more like global thermonuclear war -- everyone can play, but nobody really wins.

Come on... give it a go!

Head over to –

[HTTP://retrochallenge.net](http://retrochallenge.net)

For more info!



REVIEWS

48/128

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 3o'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 There are hidden sections to the game that can only be accessed with certain items.

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



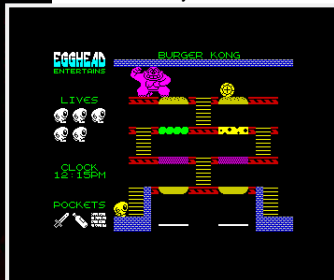
featuring a caricature of him in his very own screen. Hmmm – maybe a spin-off game featuring yours truly?

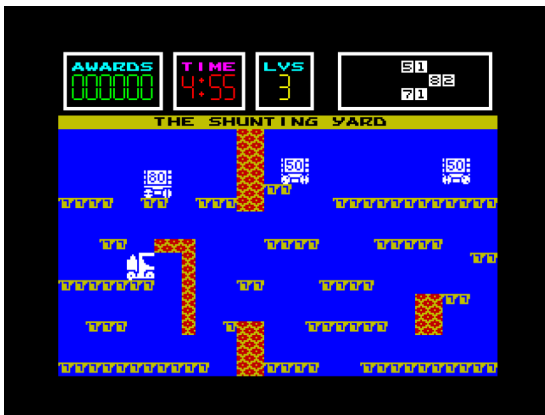
Any gripes? Well, once you get into the game, it seems to be quite small. That's more of a testament to the quality and playability of the game though. Maybe an option on the menu to set the number of ingredients required? By having to collect them all you have a potentially difficult game. Also, the 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/jonathan6egghead/id7.html)
<http://members.fortunecity.com/jonathan6egghead/id7.html>





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 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

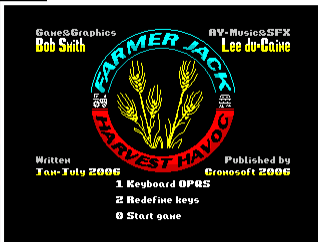
Melissa: You are Farmer Jack. A man who does love his home brewed wine! From a small hobby Jack has turned his passion for home brew into a fully fledged business. A business that just gets bigger and bigger every year until now, finally, his quality organic produce is number one throughout the land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



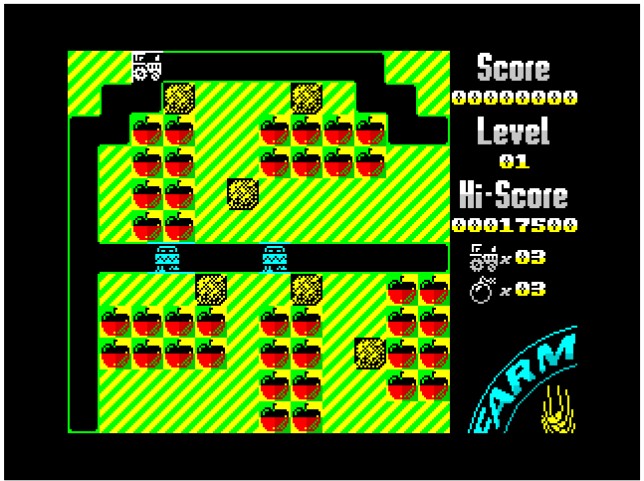
Naturally an operation like this will draw a great deal of attention. And not all of it good!

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!



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 your 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 'Dig 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?

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 bales?) 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!)

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.

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.

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%

The ZX Spectrum Book - 1982 to 199x

Author: Andrew Rollings

Some wise person once said, "Never judge a book by its cover." Sane advice when it comes to books, no doubt, but when it comes to "The ZX Spectrum Book" the cover pretty much tells you what you can expect from the book - polish, style, quality and staggering

attention to detail. Why, even cover pretty much tells you what the envelope that the book ships in has a customised Spectrum labelling on it! Beat that!

But a bit of background on the book before I give out more details on it. It was in mid-2005 (I think, well it was some time ago at any rate!) that Andrew Rollings announced he was doing a memorabilia book that covered a fair bit of Spectrum gaming history. He figured that he would finish the book by September 2005 (yes, that's not a spelling mistake). "snigger"

Of course, he didn't! In fact, it took more than a year to finish the book. And now that we have the book, we know why! The book chronicles more than 230 games covering the lifespan of the Speccy gaming industry from 1982 to 1992. Each year is colour coded to help you quickly find a relevant game. Apart from a Contents section that lists the games by their year of release, there is a handy Index section that lists the game in alphabetical order.





The entire meaty 250 odd paged book is printed in full-colour (Spectrum colours in fact) and the paper, colour reproduction and print quality is outstanding.

To cap it the foreword is written by Sir Clive Sinclair himself.

If that doesn't impress you the contents of the book certainly will.

Effectively, there are 9 chapters in the book each representing the year of interest from 1982 to 199x. Nick Humphries (of YSRNR fame) introduces each chapter with an excellent summation of the important milestones in that year. This sets a nice background and tone for the games being presented in the chapter.

Each write-up of a game is broken up into three sections that give you a brief background about the game, some trivia and a succinct overview of the game itself. The write-ups are accompanied by two in-game screenshots, the loading screen and the inlay of the game. In addition, where possible, the ratings given by the Big Three (Crash, Your Sinclair, and

Sinclair User) are reproduced to give you a fair idea of the overall perception of the game (by the magazines that is) when it came out.

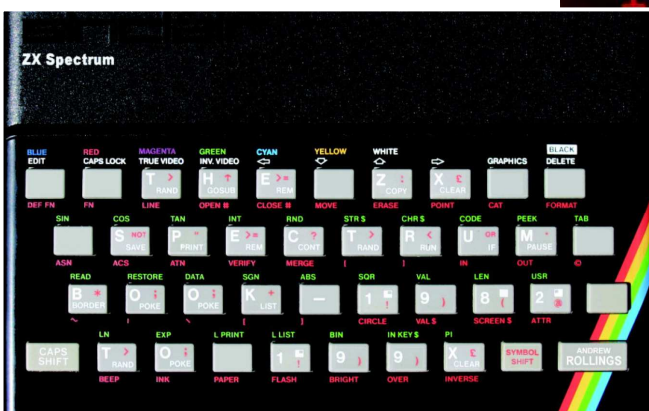
If you're wondering what kind of games are chronicled, it's interesting to note that Andrew has been rather eclectic in his selection. He writes about the good games, the bad games and even the downright ugly ones. It's a fair representation of the gaming scene that existed during that particular period. More importantly, the chosen games either have some sort of historical significance or interesting trivia that helped set them apart from the others. That's not to say he's managed to write about every interesting game that

existed – that would have been well near impossible considering the number of games that were released – but let's say every game included in the book has a story to tell and make a very interesting read.

There's nothing much else to say really. This book deserves to be on every specchums coffee table – it's a piece of Spectrum computing history (the book that is, not the coffee table). If you haven't got it already, go on spoil yourself – you know you want to!

Ratings (meaningless, I know)

- Presentation - 10**
- Content/Writing - 10**
- Print Quality - 10**
- Value For Money - 10**
- Overall - 100%**



Sinclair ZX Spectrum AllStars Volume 1

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?

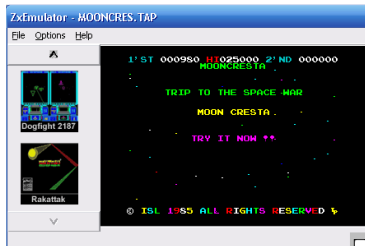
Overall, the whole package is shoddy and seems to have been rushed out. We all know

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 cannot 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%



*Published by Alten8 £4.99 (Plus 99p P&P)
Available Now*



Little Big Compo

Well, unless you've had your face down in the cat's litter tray for the last year you couldn't have failed to notice the sheer awsome and scale of Sackboy's plans to take over the entire planet! The bloody things everywhere! It's worse than the sodding Lemmings were!

Anyhoo – On the off chance you've missed the whole Little Big Boat go look it up on the internet then come back. If on the other hand you are fully aware of the sickly sweet sack creation then you will be wondering what on earth I'm doing waffling on about it in a speccy fanzine. Simples. Those chaps over at the upcoming 'Retro Reunited' event in Huddersfield this September are running a Competition.

To enter you need to design a retro gaming themed level for Media Molecules Build-Create-and-Share-em-up Little Big Planet. Once your level is complete and available to play then post in the special thread on their forum and they will all give it a play.

SO, all you Speccy PS3 owner get going! Let's see some Speccy goodness invade the Little Big Planet!!!!

For more info head over to the Retro Reunited site at

<http://retroreunited.info/>



CODE SHED

48/128

the code shed guide to BASIC – part iv

Hello and Welcome to the fourth edition of Arjun's Code Shed guide to BASIC! Now that we've created the graphics for our game and have the key handling routines in place, we can now finally move on to the fun part of animating stuff and watch the screen come alive with millions of tiny coloured pixels (okay, maybe not millions but what the heck, who's counting?)!

Firstly, if you remember the plot of our game, our intrepid hero of the game Krapz has to survive as long as he can by collecting little quanta particles until he gains enough power to jump out the warp into... yet another more difficult screen. He also has to avoid touching the "ST fluctuation trails" he leaves behind in his wake plus he must stay clear of the deadly boundaries of the warp. In order to pull this fancy (oh sure! - Ed) stuff off, we'll break it down to a set of tasks.

Task #1: Set up the playing area. This involves drawing the boundaries of the warp and populating the playing field with randomly distributed set of quanta particles. Since every new screen that Krapz jumps into involves the above steps, it makes sense to package it as a sub-routine. Which is what we'll do:

```
115 LET quanta=5*level: LET
time=seedtime+
(level*30): CLS
120 PRINT AT 1,0; INK
3;"bbbbbbbbbbbbbbbbbbbb
bbbbbbbb";AT 20,0; INK
3;"bbbbbbbbbbbbbbbbbbbb
bbbbbbbb"
130 FOR f=1 TO 19
140 PRINT AT f,0; INK 3;"b";AT f,31; INK
3;"b"
150 NEXT f
160 LET x=10: LET y=10
250 FOR f=1 TO 5*level
260 LET p=INT (RND*17)+2
270 LET q=INT (RND*28)+2
```

```
280 IF ((p=y AND q=x) OR (ATTR
(p,q)=5)) THEN GO TO 260: REM don't
overwrite player or
another quanta!
290 PRINT AT p,q; INK 5;"g"
300 NEXT f
999 RETURN
```

The first line looks a bit complex but is actually a rather simple way of increasing the difficulty level as Krapz Progresses through the game. The first **LET** statement simply increases the number of quanta particle in a screen by a factor of 5 depending on which level (screen) Krapz is playing on. For the first screen (level = 1), there will be only 5 quanta particles to collect. On the 2nd screen (level = 2) there will be 10, and so on and so forth. The second **LET** statement controls how much time Krapz has to collect all the quanta particles. This is again a factor of the level Krapz is on – 30 times the level as a matter of fact, which compensates for the fact that there are more quantas on each screen as we progress. You'll notice that a *seedtime* has been tacked on in the equation to ensure that the player has a basic minimum time to start with on each level. After having cleared the screen, it's time to draw the boundaries of the warp – in our case it's simply a rectangular arena to delineate the playing field. First we draw the top and bottom edges, which is simply a matter of printing 32 characters of the UDG "b" (incorrectly reproduced as lb in the listing above) that we created earlier on. Line 120 prints the top edge and the bottom edge in lurid purple ink.

To draw the vertical left and right edges we employ a **FORNEXT** loop from line 130 to 150 that print UDG "b" 19 times vertically. It's now time to randomly populate the playing area with quanta particles. Line 160 first sets the initial co-ordinates of Krapz on each screen – he always starts on row 10, column 10 on every screen.

We then employ a **FOR-NEXT** loop to print the actual quanta. The number of particles is simply a factor of the level the player is on, and is in fact exactly the same as the variable *quanta* we calculated on line 115. In fact we can substitute the *5*level* with *quanta* and the loop will work just as well. Line 260 and 270 calculate the row and column (p,q) to print the quanta on. *INT (RND* 17)* gives us a random number from 0 to 16. We add a factor of 2 to ensure we don't print anything on row 0 and 1 – the top edge of the playing arena. Note that if got 16 and added 2 we get 18, which still within the bottom edge (column 20) of the arena. Similarly, we keep the quanta within the left and right edges of the playing arena in line 270. Line 280 introduces a necessary check that ensures that we aren't printing a quanta on top of another already existing quanta on screen (we do want all our quanta's visible individually on the screen!). Plus we don't want to print at the position Krapz is already in (10,10). So, what we are checking for is "whether the quanta co-ordinates are the same as Krapz's co-ordinates OR whether the quanta particle is being printed on top of another quanta" state. This is achieved by the **IF** statement, where the two expressions are separated by an **OR** (as in the above statement) condition which returns true if any one of the expression is true (either printing on Krapz or printing on a quanta).

To check for whether the quanta co-ordinates are same as Krapz's co-ordinates, we simply have to check if the column and row co-ordinates for both match up. If they are the same, then we are at the same location on screen. This is calculated using the expression in the **IF** statement $p=y$ **AND** $q=x$. The **AND** condition returns true if and only if both the subexpressions are true.

The next thing we have to check for is the case of a quanta overwrite condition. This is done by using the **ATTR** command, which given a x & y co-ordinate returns the colour attribute at that co-ordinate. The BASIC manual explains how to interpret this value but suffice to say that for our purpose if **ATTR** returns a value of 5 it means that a cyan coloured character (INK 5) is present at that position. As you will see on line 290, where we print our quanta particles, they are printed with INK 5. In essence, what we are saying is that if **ATTR** returns 5, we assume that we have a quanta particle there regardless of what actually *might* be there. In order to not trip up our assumptions we ensure that we do **not** print any other stuff in INK 5 anywhere the screen. In our program only the quanta particles can have the cyan colour when playing the game. Period.

If we are overwriting Krapz or another quanta, we simply go back to line 160 and re-calculate a new quanta position until we are satisfied we aren't overwriting anything we shouldn't be overwriting! If all is well, we proceed to actually print a quanta particle on the screen at the position we calculated (line 290). Once we've printed all the quanta particles required for a level we **RETURN** from the sub-routine via line 999.

There! We're all set to actually do some game play and stuff!

Let's move on to the task of moving Krapz around on the playing area. Remember we set up our control keys in Code Shed Guide 2, so it's time to use those keys to move Krapz around. The control scheme we'll follow is a simple one. Pressing one of the four direction keys changes Krapz's direction instantly (inertia? What's that?). The key needn't be held down if one wishes Krapz to continue in that direction though- his inertia (that word again!) will keep him going in that direction until a different direction key is pressed. Have a look at this code which begins our main game loop (so called because we'll be executing it repeatedly till something happens to break proceedings) proper:


```

1010 PRINT AT 21,0;"Lives: ";lives;TAB
20;"Score: ";score
1020 PRINT AT 0,0;"Time: ";time;" "
1030 PRINT AT 0,20;"Level: ";level
2000 LET a$=INKEY$
2010 IF a$=k$(1) THEN LET dir=1:
REM right
2020 IF a$=k$(2) THEN LET dir=2:
REM left
2030 IF a$=k$(3) THEN LET dir=3:
REM up
2040 IF a$=k$(4) THEN LET dir=4:
REM down
2045 IF dir<>0 THEN PRINT AT y,x; INK
1;CHR$(143)
2050 IF dir=1 THEN LET x=x+1
2060 IF dir=2 THEN LET x=x-1
2070 IF dir=3 THEN LET y=y-1
2080 IF dir=4 THEN LET y=y+1

```

Lines 1010 to 1030 give information regarding the status of the number of lives left, the current score, the time left and the current level being played. It's our HUD if you will. Since it's within the main game loop, it will be updated continuously. Lines 2000-2080 are responsible for our inertial control mechanism. The logic is quite simple.

We sample the keyboard for any key press. If it's a direction key we change Krapz's direction of movement. If no directional key is pressed, Krapz continues to move along in the same direction. Here's how.

Line 2000 reads in a key from the keyboard (null if no key is pressed). Lines 2010 to 2040 compare the value in a\$ with the value in k\$ (our desired control keys). Depending on which key is pressed, a variable called dir is set to a particular value that signifies a direction. Line 2045 prints the trail behind Krapz. Basically, all it does is see if Krapz is moving (Krapz doesn't start moving until you press a key initially), we don't bother printing a trail. If he is moving (dir will have some non-zero value then), we will print a solid block of blue at

the current Krapz co-ordinates. **CHR\$(143)** will print the solid graphic block you see on the numeric 8 key on the Speccy keyboard (Graphic mode + shift + 8). What's this **CHR\$(65)** you ask? Well it's one way of printing a character on the screen. For example, if you do *PRINT CHR\$(65)* it will print the letter "A" on the screen. This is because **CHR\$(65)** converts a number to its string equivalent from the character set. If you take a look at the character set table in the Speccy manual you'll see that numbers from 32 to 127 represent characters from the ASCII standard. Which is why 65 corresponds to letter "A". From 128 to 255, the character set is unique to the Spectrum with some special characters like the Graphic Blocks taking up positions 129 to 143. You can print any of these characters by passing the code number to **CHR\$(f)**. Some unprintable characters (no, not the sort you're thinking of) can do some fancy print formatting tricks – refer to the manual for more on that.

Coming back to the code, so we print the blue block at the current Krapz co-ordinates to signify a trail. "Hang on!" you say. "Won't we be overwriting Krapz in the process?". But of course! But since we'll be repositioning Krapz at new co-ordinates anyway, it doesn't matter. In fact, the code to calculate Krapz's new co-ordinates follows:

```

2050 IF dir=1 THEN LET x=x+1
2060 IF dir=2 THEN LET x=x-1
2070 IF dir=3 THEN LET y=y-1
2080 IF dir=4 THEN LET y=y+1
2090 IF (ATTR (y,x)=3 OR ATTR (y,x)=1)
THEN FOR f=0 TO 4: PRINT AT y,x;
PAPER 2; INK 6; FLASH 1;CHR$(145+f);
FLASH 0: BEEP 0.4,RND*f: NEXT f:
PAUSE 10: GO TO 6000

```

Lines 2050 to 2080 recalculate the new x or y position depending on the direction of travel. Simple, eh? In line 2090 we check whether Krapz has crashed into the boundary walls or walked into his own trail (ST fluctuations are bad remember?). This is handled by a single **IF**

statement that checks whether the attribute colour at the newly calculated Krapz co-ordinate matches **INK 3** (the magenta colour with which the walls are drawn) or **INK 1** (blue colour of Krapz's trail). If it is, it's time to play a sad ditty expressing our condolences and create a nifty looking explosion. Since I'm not musically inclined I'm just using a simple **BEEP** statement that randomly plays some stuff in a low pitch. Creating the explosion is simplicity in itself – I've created 4 explosion UDG's using Basin's UDG creator tool. To simulate the explosion effect, I just print the four characters (with full **FLASH** and stuff for added effect) one atop the other with a small time gap inbetween to slow things down a bit. It's not the greatest explosion effect you'll ever see but it's good enough for our purpose. After the dust has settled down, we send the program off to line 6000 where the last recitals are performed. More on that later.

```
2095 IF ATTR (y,x)=5 THEN LET
score=score+(level*10): LET quanta=quanta-
1: BEEP 0.01,0.1: IF quanta=0 THEN
PRINT AT 10,3: PAPER 1: INK 5: FLASH
1;"Space-Time Jump! Get Ready!"; FLASH
0: FOR f=1 TO 10: BEEP 0.03,RND*f:
NEXT f: PAUSE 50: LET level=level+1: GO
TO 6020
3000 PRINT AT y,x: INK 4;"a"
```

Line 2095 determines what happens when Krapz picks up a quanta particle (we simply perform an attribute check for that). What happens is this: first the score is increased by 10 times the level we are on (as a measure of difficulty). The number of active quanta's on the screen is decreased by one. If there are no more quanta's left, it's time to trigger a Space-Time jump and warp to the next level which is easily done with a simple message, a few random beeps and a jump to Line 6020 that sets a few variables back to initial values and then re-draws the level screen by jumping to line 1010. If a space-time warp hasn't been triggered (because there are quanta's still left on the

screen), we go on to print Krapz at the new location in line 3000. We come to a final bit of code that deals with the player running out of time.

```
3005 LET time=time-1
3010 IF time=0 THEN PRINT AT
10,5:PAPER 2: INK 6: FLASH 1;"S-T Field
Collapsing!"; FLASH 0: FOR f=10 TO 1
STEP -1: BEEP 0.1,f/2: NEXT f: PAUSE 50:
FOR f=2 TO 19: PRINT AT f,1: PAPER 1:
INK 2;"::::::::::::::::::::::::::::::::::
::::::::::::": BEEP .05,RND*f/2: NEXT f: GO
TO 6000
4000 GO TO 1010
6000 LET lives=lives-1
6010 IF lives=0 THEN PRINT AT 10,10:
PAPER 4:INK 1;" GAME OVER! ": PAUSE
100
6012 IF lives=0 THEN IF score>hiscore
THEN LET hiscore=score: PRINT AT 12,10:
PAPER 5: INK 1: FLASH 1;"New High
Score!"; FLASH 0: PAUSE 100
6015 IF lives=0 THEN RETURN
```

Line 3005 simply decreases the available time by 1 unit. Line 3010 determines what happens next. If we've run out of time it's time for another dramatic message with annoying random beeps and appropriately dangerous looking flashing text. We up the ante by printing a line of solid block (the odd `:A` is the same GRAPHIC MODE + SHIFT + 8 friend we met a while back) that slowly fills up the entire screen to signify a field collapse. We then jump to line 6000 to finish off the formalities. If we haven't run out of time we simply loop back to line 1010 via Line 4000 to continue playing. Line 6000 is where the case of death is handled. First the number of available lives is decreased by 1. Line 6010 checks if we've used up all our lives, in which case it's really Game Over! Line 6012 next checks if we created a new high score by simply comparing the current score with the high score. If our current score is higher, we save it as the high score and congratulate the player on

achieving the same. Line 6015 proceeds to return from the main game loop routine and dumps us back at the main menu screen from where we'd first come from thus bringing us back full circle to square one.

Tying up the loose ends

That's the majority of the game code out of the way. That justleaves us with tying up some loose ends:

```
110 LET lives=3: LET dir=0: LET score=0:
LET seedtime=50: LET level=1
```

This is part of the initialization phase where we set up the game. We set the number of total lives to 3, set the current direction to 0 (not moving), score to zero and the seedtime (that we discussed earlier) to 50 and the current level to 1. This is done each time we start a new game from the main menu. If you're wondering when line 110 is called, it is done so when we enter the main game loop the first time here:

1005 GO SUB 100

In the main game loop we must also remember to reset the start state for Krapz for every screen. This is done here:

```
6020 LET dir=0: LET x=10: LET y=10: LET
a$="";
PAUSE 100: GO SUB 115: GO TO 1010
6050 PAUSE 0: RETURN
```

Line 6020 basically resets the co-ordinates of Krapz to the center of the screen (relatively), sets *dir* to zero, and calls line 115 to seed the quantas and timer value before finally setting up the new screen with a call to line 1010.

Finally, we print out the instructions to play the game from lines 9000 to 9070:

```
9000 REM *****Main Menu*****
9005 BORDER 0: PAPER 0: INK 6: CLS
9010 PRINT "AAAAAAAAAAAAAAAADash
it!AAAAAAAAAAAAAAAA"
9015 PRINT
```

```
9020 PRINT "Krapz the mysterious alien
finds himself trapped in a Space-Time
continuum warp. The only way to survive is
to keep on the move, and collect the quanta
particles in the vicinity that will give him the
necessary boost to jump out the warp."
```

```
9025 PRINT
```

```
9030 PRINT "Be forewarned though that
there are other warps in the vicinity too and
you may well jump into another more tricky
one."
```

```
9035 PRINT
```

```
9040 PRINT "You must also stay clear of the
ST fluctuations you leave behind in your
wake. Touching them will mean instant
death. Avoid the walls of the warp too for the
same reasons."
```

```
9050 PRINT #0;"Keys: Left=";k$(2);"
```

```
Right=";k$(1);" Up=";k$(3);" Down=";k$(4)
```

```
9060 PAUSE 0
```

```
9070 RETURN
```

The listing is fairly self-explanatory but to reiterate, lines 9020 to 9040 print out the actual instructions. Line 9050 also helpfully list out the control keys being currently used in the game. Note that the *f* in the line 9010 is actually the UDG "f" character. And that my friends brings us to an end to the Code Shed Guide to writing a BASIC game! Hopefully, you'll go on to write better and bigger stuff than the Dash-it game described here.

LOST TREASURES

Welcome to yet another instalment of Ian Hamilton's *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 rereleased 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.

PROGRAMMERS Diary

This issues diary follows Jonathon Cauldwell's journey through 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 voiding 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.



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.

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.

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.

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 character. Chug Horncastle has a nice ring to it.

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...



GOING COMMANDO

Continuing from last issue, Mr. Anonymous continues his journey through Command to show us how to make our own levels.

\$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
defb $82, $9A ;add more space.
                ;draw a cliff in
                ;column $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 jump
                ;off
```



```
efb $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)
```

GOING COMMANDO

\$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:

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

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

\$95 Barracks

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



GOING COMMANDO

\$00, \$13 ;insert more space.
This value will ALWAYS
be the same
\$1D, \$1A ;finally add the top
cave section

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
```

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.

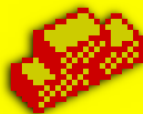
Example (Without Motorbike)

```
$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
```

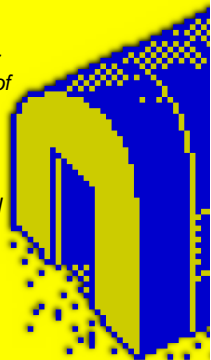
Example (With Motorbike)

```
$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
position $3C ;(approximately
half way across screen)
$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
```



GOING COMMANDO



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
```

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>
```

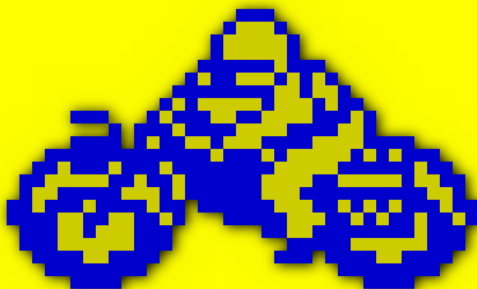
```
$28 , $1E
```

```
<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.



GOING COMMANDO

\$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.
Roads run diagonally down from the left

```
$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
```

Join us next issue where we conclude the article. If you are nice to use we will even include a reference guide!



INTERVIEW

48/128

Richard Jordan of Retrospec has kindly agreed to answer a few questions regarding the recently released Atic Atac. Ben Rapier wears the interviewers hat.

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 shuffy 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 dam 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.

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: 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 nonstop then I reckon we could have got it all done in something like 8 weeks. A bit of a guess

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

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 Underwurld.



We'd all like to give Richard and Ben somewhat belated thanks for their time. Please be sure to visit the Retrospec website at –

<http://retrospec.sgn.net/>

Now, make sure you're sting comfortably for this? Settled? Good. Next month 'The Gaffer' interviews none other than 8 bit footy legend KEVIN TOMS!

So make sure you get your next issue of ZX Shed Reloaded, out 1st Friday of the month!



SPECIAL FEATURE

THIEVES SCHOOL

ESCUELA DE LADRONES

Carl - When you think about it, computer games are a con. I mean, you play a game in which you collect a ton of gold coins, yet at the end of it you have no more real wealth than when you started. Take today for example, with all the jewels and priceless paintings and stuff I've stolen in the game Thieves School, I should be richer than Russell Brand's therapist. Instead I'm as poor as an unemployed church mouse who's just had his benefit stopped (and at Christmas too!). All in all, then, it's a good thing that Thieves School is so enjoyable, as it won't advance your wealth by a penny.

Thieves School is a 128K only game by World XXI Soft Inc. No, I've never heard of them either, but judging by Thieves School they're well worth looking out for. In Thieves School you play the part of a trainee thief taking part in a final exam. Not (thankfully) a written exam, but a practical - you get to go around the world and steal lots of things from museums and galleries and stuff. Wish I'd gone to a school like that.

THIEVES SCHOOL

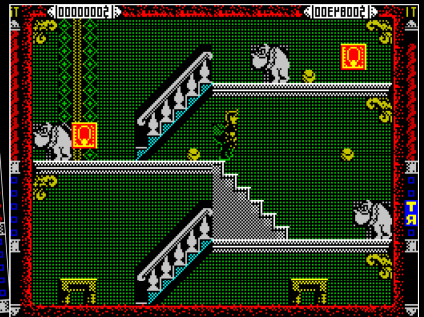
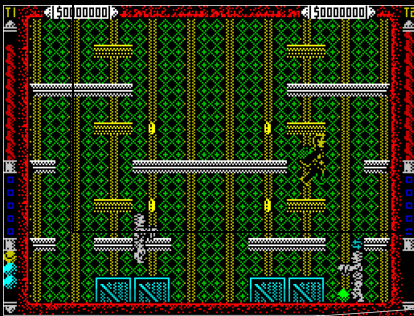


START
CONTROLS
SOUND
ESPAÑOL
EXPANSION
CREDITS



You control either a male thief called "Courage" (perhaps a joke name in the author's first language of Spanish, I don't know) or a female thief called "Mist", or if you play co-operatively (yes, you can in this game!) then each player chooses who to play. Courage is stronger than Mist, but Mist is a better lock picker. See, you choose whether to play this game in one of two modes; Action-Robbery and Stealth-Larceny. The latter mode plays like a traditional platform game, in that if you touch a moving enemy then you lose a life, and your only defence is to avoid the roaming enemies. In the former mode (Action-Robbery, in case you can't remember two sentences back, and if you can't then you've probably forgotten

that you owe me a tenner, send it to me now, before you forget) you still must avoid touching the moving enemies as they kill you on touch, but you do have a defence; you can throw red and green apples (that's what they look like) and stuff at them. You have to pick up the objects before you can throw them, and when you throw them they follow a sort of laws of physics type throw / bounce / fall pattern. Alright, so the physics don't beat those in Half-Life 2 but it's still a nice touch. And this game runs on a Speccy, which is a much nicer computer than a PC, which you need to play Half-Life 2. Anyway, if you play in Action-Robbery mode you get to throw things at the enemies and stun or kill them -



different enemies take different number of hits to kill them, until that number is reached each hit temporarily stuns them.

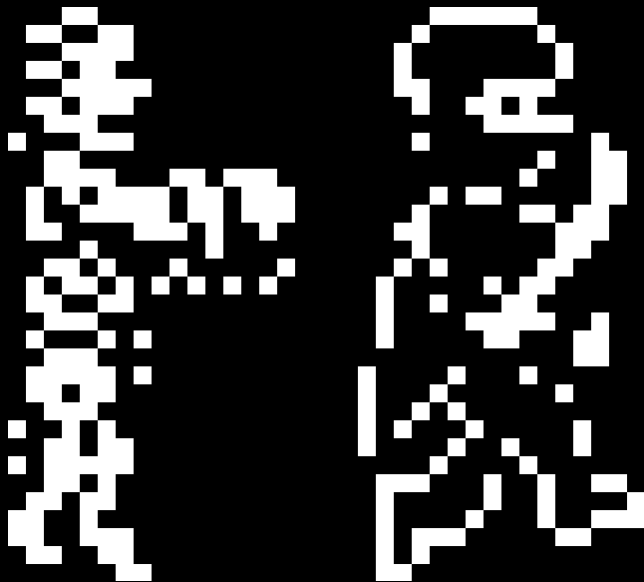
You are first introduced to Courage and Mist via side A of the tape. Or the tape image of Side A. See, when you buy Thieves School you get a .zip file (currently no cassette version is available, but hopefully that will change for the true Specy fans) that contains the instructions in English and Spanish, two printable graphic images of the cassette inlay (so you can make your own cassette version of the game), and four game files. The first game file is a .tap version of side A of the tape. This is the introduction to the game. The second game file is a .tap file

of side B of the tape. The third and fourth files are .tzx versions of sides A and B, for which the authors are to be commended; offering both .tap and .tzx versions of the game is an excellent idea, as many people love the real loading style of .tzx, but others (like me) prefer the ease and speed of .tap files. So full marks there then. It even makes up for them not having an apostrophe in the name Thieves School. As I said earlier (if you were paying attention, if not then I'm not going to repeat myself) you are first introduced to Courage and Mist via side A of the tape. When you load side A, you get an animated intro showing the story (such as it is) behind the game. The graphics are not the best around, but still rather good. And though you

probably won't watch it more than once, it's worth seeing it through, so you know what's going on in the game. Or you could just live in ignorance, which is good practise if you want to work in PC world, I suppose.

Anyway, onto the game. After you've loaded it (128K only, so no rubber keys here, sadly) you get a menu along with a picture of Natalie Imbruglia (I think) and some bloke who looks like me, only not so good looking. Or it could be Pierce Brosnan pretending to be me, I don't know. I wouldn't put it past him. Anyway, between him and Natalie there's a menu where you can choose to start the game, define the controls, alter the sound, load expansions, and view the game's credits. Yep, expansions. See, the game is programmed in such a way that you can load expansions, which are things like new rooms, new game modes and so on. It remains to be seen what will appear, and I do wish that a screen designer had been included with the game, but it's a nice feature and hopefully will enhance the game's longevity for a long time to come.

When you start the game, the graphics are the first thing to strike you. They really are rather nice, with great use of colour and backgrounds (not too usual in Spectrum platformers, instead we're more used to seeing an all black background). The character



ESCUELA DE LADRONES

your control isn't too large, being one character square wide and two tall, but he (or she, if you play Mist) moves well, with no response problems or iffy collision detection.

The sound too is very nice, and the whole thing looks and sounds very pleasing, and is obviously programmed by someone who knows their way around the Speccy. Even more impressively, despite the really nice use of colour, there is no noticeable colour clash (unless you look for it, and if you do, you really need a hobby. Might I suggest fact spotting. This is where you go on a C64 forum and see how quickly you can find a true fact. The current record is about three days). The game even has a high score table to round things off. Although why do games persist in making you use UP and DOWN to select the letter to be entered, when you have a perfectly good keyboard at hand? It's a Spectrum, it comes with a keyboard. Which brings to me another minor but annoying aspect of the game; to select an option in the menus (Play Game, Redefine Keys, etc) you must press 6 to move the selection "bar" down, 7 to move it up, and press ENTER to select it. Even if you've redefined the keys to be Q, A, O, P and SPACE, or whatever. Why? In the high score name entry screen you use the defined keys for up, down and fire, so why not on the menu screen? Or rather, why doesn't the menu screen recognise the keys 6, 7, and ENTER, *and* the currently defined keys.

Still those are little niggles, and not exactly game killers. And in just about every other way, the game is very well polished and thought out indeed. And playing the game is a lot of fun too, at least in the Action-

Robbery mode, where you can stun/kill your enemies with well timed throws. There are lots of screens, you can control your character in mid-air, and you get lots of lives to play with. All well and good, but sadly nothing's perfect (well, apart from Irm Bru and chips, of course) and Thieves School's only real flaw is in it's other mode; Stealth-Larceny. See, in Stealth-Larceny mode, not only can you not stun or kill the creatures, but you have to pick locks too. The problem is that

picking locks seems to be a matter of luck - you have to press fire when your character is over a safe to active lock pick mode, then you are presented with a small graphic of the interior of the lock, and you have to "pick" the lock by pressing fire when the tumbler is up, allowing the lock pick through. You must do this for every tumbler, and if you get one wrong then the lock pick fails, and you have to start again. The trouble being that the tumblers move so fast that it's more or less down to luck if you succeed, and since you can't stop the enemies, they can kill you whilst you are (repeatedly, and I mean repeatedly) trying to pick the lock. To me, this really hurts the Stealth-Larceny's modes



THIEVES SCHOOL



Thieves School is for the 128K Spectrum (not 48K, sorry), costs \$5 (around £3.50) and is available from



<http://ws.vtrbandaancha.net/>

playability, which is a real pity. I can easily imagine that the author will end up releasing a POKE to slow down the speed of the tumblers.

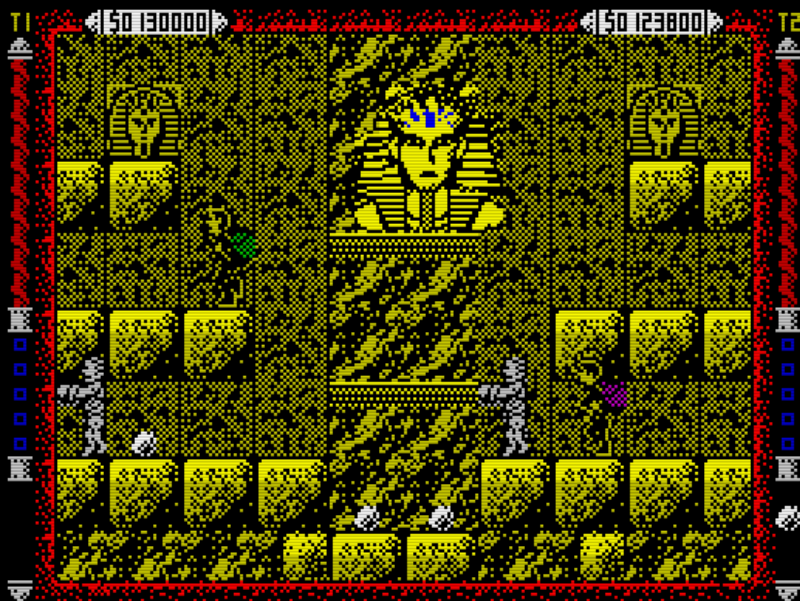
Still, some people might like the Stealth-Larceny mode regardless of the too-fast-for-comfort lock picking, and even if not, the Action-Robbery mode of the game is really very enjoyable. It's a pity

that the game is a little repetitive, and that there's no skill level (which would have helped the lock pick bit no end!) but it's still a very nice game. Not a mega-game, unfortunately, but well worth buying (and the author really deserves the cash for all the skill and effort he's put into this game). And I'm really looking forward to the next game by this talented author.

and you can also download more games (48K and 128K) for free from that same website.

Marks
Playability - 8
Graphics - 9
Sound - 9
Instant Appeal - 8
Addictiveness - 7

For next month special feature Sard provides us with a remake special! Groovy...



PAPERBOY



The Best Names in Mobile Games www.elite-systems.co.uk

Elite Systems Ltd, 10-12 Chase Road, WS7 0DP, England Email: enquiries@elite-systems.co.uk

Paperboy™ & © Midway Games West Inc 1984-2004. All rights reserved. Used under licence by Elite Systems Ltd.

Jupiter ACE

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.

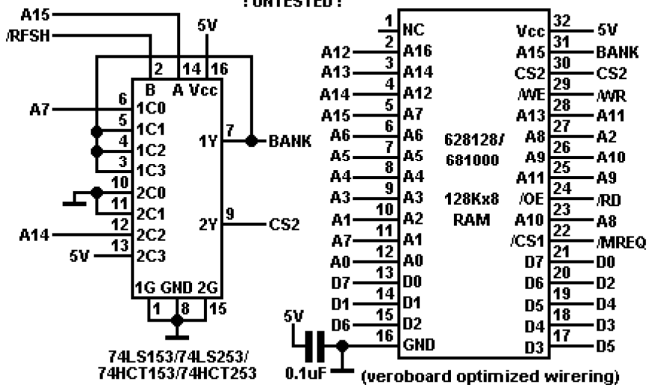
A new section on the site has been started called customised Aces. A few JA owners over the years added modifications to their JA's we start with details of three Aces with another four to come later just as soon as I put the web pages together.

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.

Jupiter Ace 96K RAM expansion

by Edwin Blink September 2005



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.

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,
```

We will be looking at these words soon. With the EB RAMpack now debugged and working correctly we have added 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.

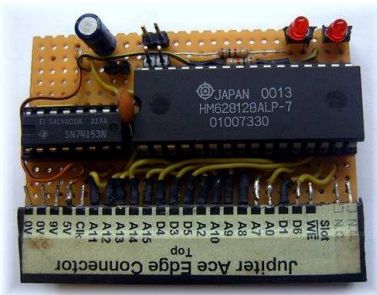
ACE Forth

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.

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 to reserve a few bytes of RAM which crosses the page boundaries.



The 96K prototype

As you can see in the above image, LEDs 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.



Jupiter Ace ACF Version 4

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 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 within 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 it's 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.





PREVIEWS

Miles Mad Mission!

"Fantasy world Miles. A new game by dm_boozefreek. Set in the world of Jetset Willy, Miles must navigate the 50 rooms of the WOS mansion collecting all the empty bottles before Beanz will let him go to bed." (original post by beanz - 6th June 2008)



And that's how it began, a throwaway remark on the forums just over a year ago thats snowballed out of control. dm boozefreek's *Jet Set Willy* 'mock up' he did for a laugh was now being turned into a full blown game that would (Hopefully) be in our hands by Christmas.

He's teased us with screengrab after screengrab, joke after joke, pun after pun as the project plodded along...

And from the looks of the screens we've been treated to, this game is going to be something very special indeed.

The game is being written on the 'JSWED' editor (Available



in the WOS archive) and most WOS regulars seem to be turning up in one form or another within the screens... and sitting on the edge of their seats in anticipation of a final release. Sadly the game missed the proposed Christmas release but to ease the pain we did get a demo, which was pure awesomeness in every sense of the word. **Miles Mad Mission** should hopefully be with us by the Autumn.

Expect a full review the second we get our grubby little mitts on it. In the meantime you can check the games progress in the 'Games' forum under '**The amazing Miles Mad Mission thread!**'



Miles Mad Mission!

After a particularly rowdy night at the WOS mansion things got slightly out of hand, the party went mental, Grizzly murdered a Lemonhead who was posing as a WoSser, Foggies missus turned up and made him go home just as he was about to stab somebody, TKrap finally built the ultimate uber specky out of tinfoil, pritt stik, and cherry liquorice laces, and aliens landed and took Cornish Davey to planet blisstonia where he took his MIA's.

Miles must navigate the WOS mansion collecting all the Specky related goodies, empty beer bottles, used condoms, empty ferrets, C90's, and more, before Beanz will let him go to bed.

Will there be "A Troll" hiding under Beanz's confusing metal bridge? Will Miles find Mels pom stash? Will B00mZilla finally show his face (or pay for the XBOX Bleugh sent him years ago). Will a trip to planet Blisstonia reveal long lost copies of Hands of Stone and the Specky conversion of Rimrunner?

.....and never, never, underestimate the power of THUNDERSTORMS!!!!



SLUBBERDEGULLION

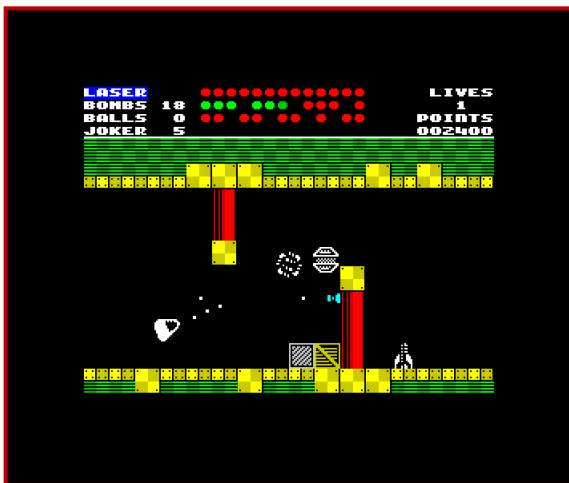
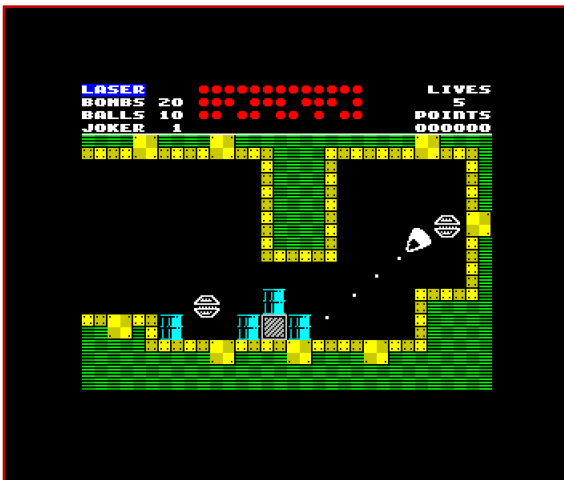
ZX SHED
EXCLUSIVE
PREVIEW

KARINGAL: We're all running out of ways to describe Jonathan Cauldwells relentless output, not even the Oliver Twins were as prolific as this man, I swear the guy must dream in Z80. No sooner does he complete a game and he's on to the next, this boy really needs to get a life. But until then we all get to bask in the glory of his latest offering.

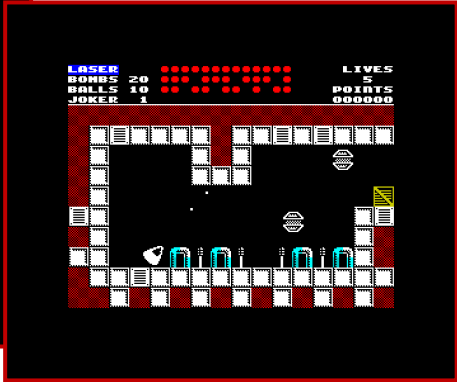
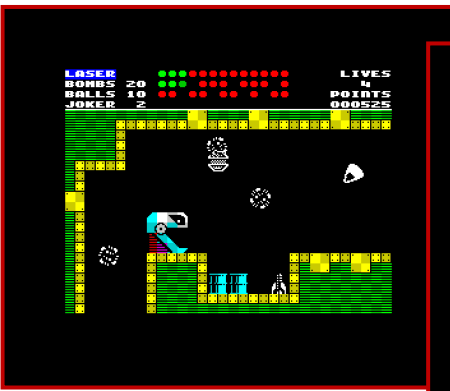
Slubberdegullion (whatever that means, if you're sad enough Google it) is said offering and without trying to

sound like a broken record definitely up to his consistently high standards. This time we get a good old fashioned shoot 'em up, imagine the bastard offspring of Asteroids and Scramble and you won't go far wrong.

Work your way through the tunnels destroying everything in sight and collect the goodies left behind. A case of disengage brain and engage trigger finger. The ship is guided Asteroids style with left and right rotation and a forward thrust. The controls get a little getting used to but it won't take long to suss them out and it'll be second nature..You get 3 types of weapons, the traditional lasers, drop bombs and scatter bombs to wreak havoc with as the aliens pop out of nowhere to mount their assaults on you so need to be ready at all times.



LOUD SPEAKER
C.82
C.55
C.56
C.44
C.8
C.7
C.6
C.5
S.P.C.168



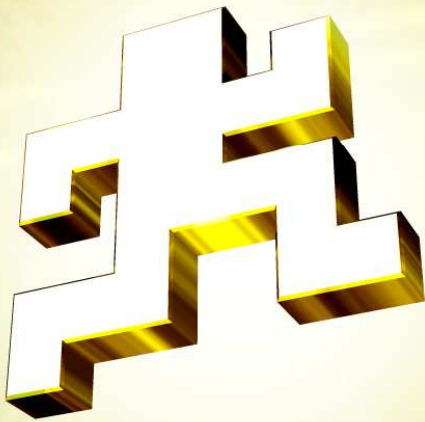
As is fast becoming Jonathans trademark, the sprite animation is silky smooth and the screens very colourful, though the sound (in this preview version at least) was limited to weapon firing and explosions and all very similar at that. Perhaps the release

version will be more varied. In fact the sound is really the thing that I can criticise in this game. It absolutely screams of that mysterious 'just one more go' factor which unfortunately so many games these seem to lack. How will it fair in six

months time? Well we'll just have to wait and see but for now Chronosoft have got another JC classic about to land in Simon Ulllyatt's trusty tape duplicator and I for one cannot wait.



RELOAD TWO



RELOAD TWO
4TH SEPTEMBER 2009