THE CLASSIC ADVENTURER

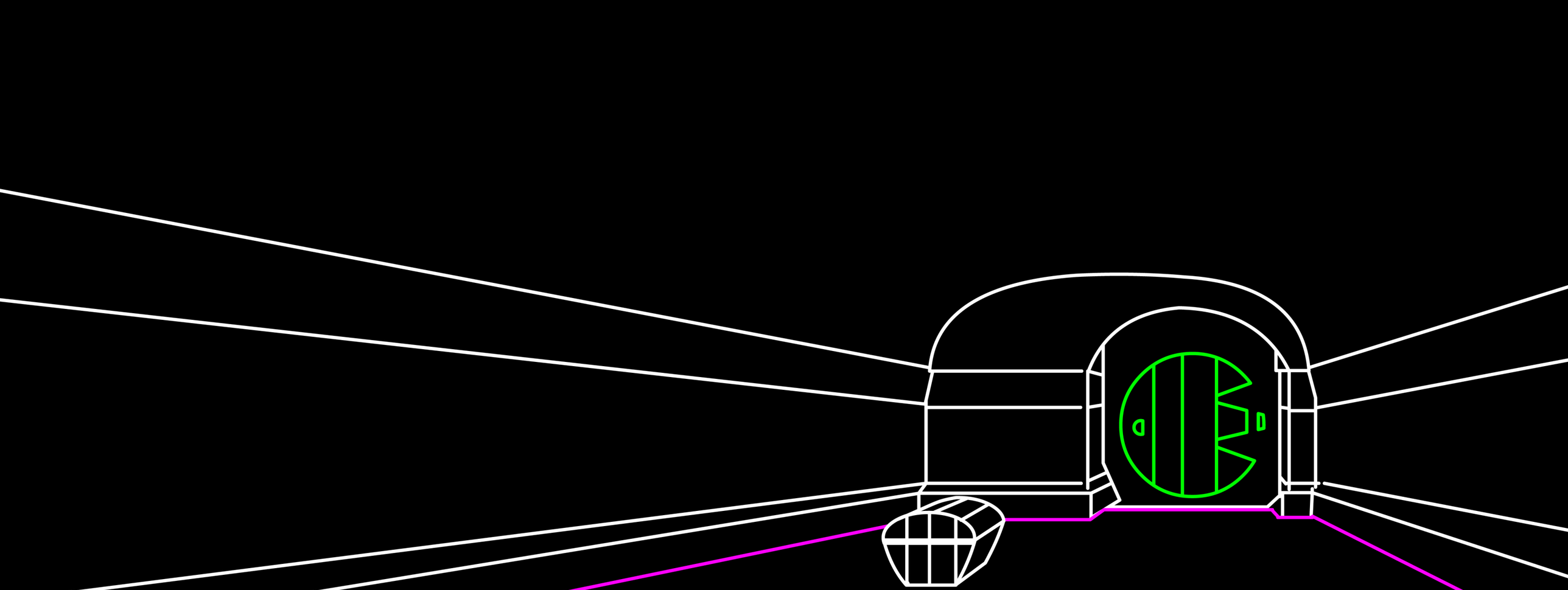
I owned an Acorn Electron as a kid. It wasn't the greatest games machine in the playground, but it did have the best game of all-time, Braben and Bell's *Elite*, and one of the finest adventure games of all time, Trevor Hall's *Twin Kingdom Valley*.

For a boy with a fertile imagination, and an obsession with the Fighting Fantasy books, *Twin Kingdom Valley* whisked me through the screen, and into a fantasy world of babbling brooks, Forests, Orcs, Trolls, Goblins, Dragons, Kings and treasure!

I played as many adventures as I could, but It wasn't until I owned a ZX Spectrum and Fergus McNeill's *The Big Sleaze* that I encountered the same immersion with another game. A friend and I spent many weekends hunched over the keyboard, notepad and pen, determined that Sam Spillade would find the missing Maltese Bullfinch.

I'm therefore delighted that both Fergus and Trevor feature in this celebration of classic adventure games, along with many other adventures and authors that transported legions of other kids to far flung corners of their own imagination.

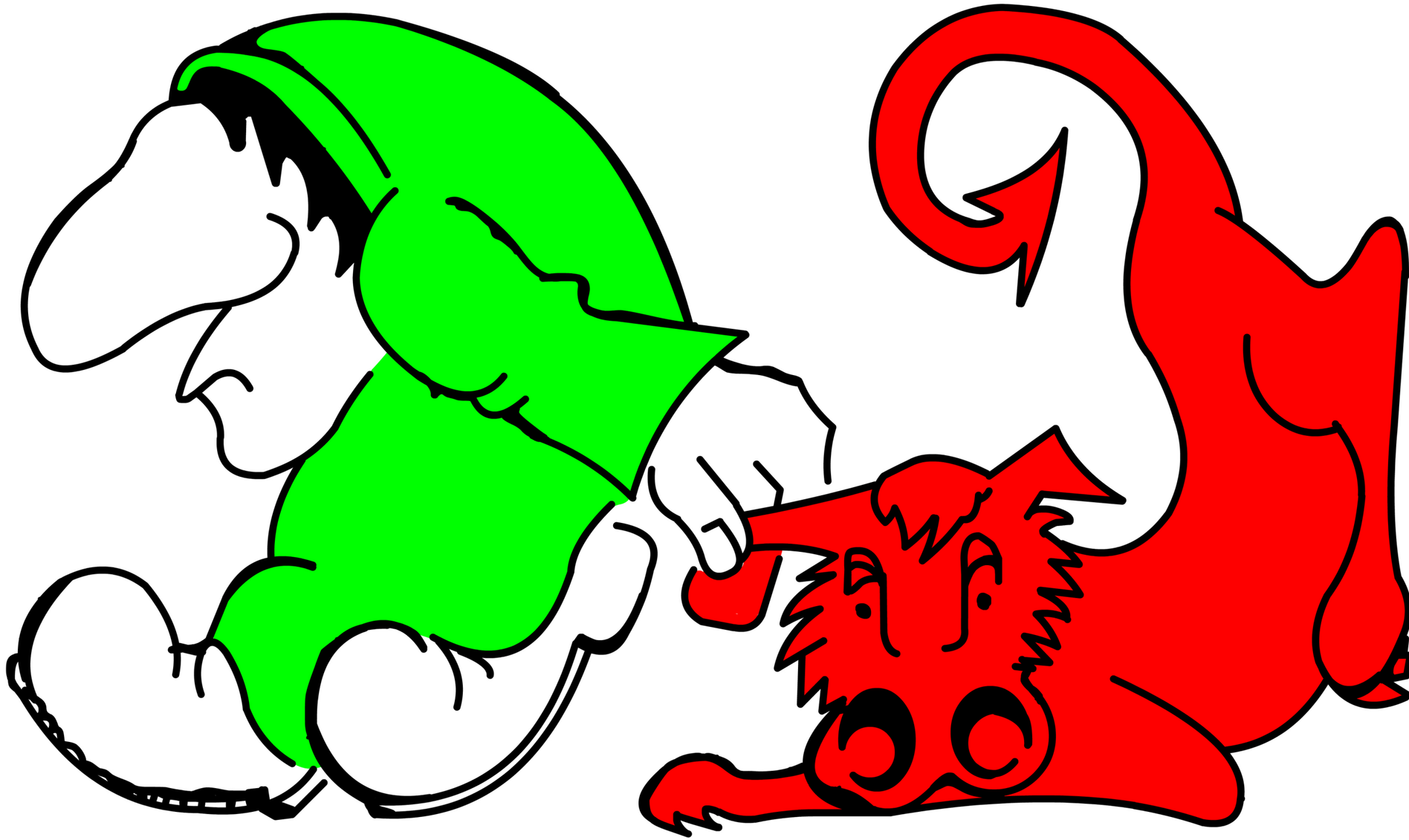
Mark James Hardisty, 2026



CONTENTS

JOHN JONES-STEELE

John Jones-Steele created *Adventure I* was one of the first and best commercially available versions of Crowther and Woods seminal *Adventure* in Britain.



John Jones-Steele's interest in computing started at any early age as a grammar school pupil in the late 60s. When his parents moved to Aberystwyth in 1971 he applied for a job as a Computer Operator at the city's University, maintaining and running their ICL 4130 mainframe. He progressed to become Shift Leader, something required as the computer ran for 24 hours a day and had to be supervised at all times. The University encouraged staff to learn programming and Jones-Steele become proficient in Assembly Language, *FORTRAN* and *COBOL*, along with a host of other languages including *PASCAL* and *ADA*.

[John] At this time I was writing mostly in the pre-cursor to C, Compiler B and BCPL. I used this to develop a small Chess program that would eventually be rewritten in Assembly and released on the Sinclair ZX81. I started playing around with the department's Apple II and Commodore Pet and when the ZX80 came out I started looking at developing games in Assembly Language.

As technological change rapidly advanced, the University replaced the original mainframe machine with a new Honeywell 6000 series in the late 70s. This came with a version of *Adventure*, or *Colossal Caves* already installed.

[John] I really loved the game and decided to write a version for the 16K ZX81 as soon as it came out. I managed to get hold of the original *FORTRAN* version while visiting the University of Manchester's computer department and proceeded to recode the data into a primitive database. That way I could see exactly how the logic worked.

The major challenge for John was figuring out how to squeeze the game down to run on the humble ZX81. The original 1976 PDP-10 version consumed around 70K, which grew to around 100K when Don Woods expanded it in 1977.

[John] I [started to] look at [how] to squeeze the game into the limited memory. I'd read up quite a few on early versions of [text compression] and worked out how they were doing it. I [ran] something on the mainframe that would scan all the data and work out ways of taking groups of letters and converting them. It would go through and find maybe 3 letter combinations that were repeated

more than any other one and then I could compress those down from 3 bytes to 1 byte. So, we were getting down to a third of the size straight away. It was a case of trying it with bigger groups of letters and seeing which was the most efficient one. [...] I tried to use one-byte codes for the biggest ones and then two-byte codes for some of the bigger words that were used multiple times.

Though he had access to machines with greater capacities than his ZX81, John realised that Sinclair were about to make a home computer game market viable with their cheap British computer. His experiments soon moved onto fully blown arcade games, and he started a small cottage industry label called Abersoft – named after his employer but with no official link to the University.

[John] Abersoft was Aberystwyth Software. I continued working full-time at the University and wrote games in my spare time, the shift work meant I had lots of spare time during the day and as soon as the ZX81 came out with the non-flashing screen I wrote a version of *Invaders*, *Pac-Man* and *Chess*. These were reasonably successful and helped fund purchasing all the other computers around at the time which I wrote versions for.

He continued to toil on his port of *Colossal Caves*, and his completed version was released in April 1982 as *Adventure I* for the expanded ZX81 computer. Contrary to several documented disassembles of the game, John developed his techniques in isolation, without reference to the influential text adventure methods first mooted by Ken Reed [featured in the August 1980 issue of Practical Computing magazine] or Scott Adam's interpreter that was written to fit games into a machine of similar capabilities. The resultant program was a compact 13K of machine code and 1.5K of BASIC.

[John] It was all my own ideas. I'd spent quite a few months with the *FORTRAN* program, ripping out all the data I could and working out how I could compress it to make it fit. The actual logic wasn't particularly difficult. [...] I developed a primitive database system that had custom codes for movement, etc. It was a format that was probably a primitive version of an *A-Code* data layout [the technology developed by the Austin Brothers at Level 9 that John would encounter later in his career]. At that time there wasn't much information written about parsing languages, so I wrote a simple system that basically allowed for a verb and a noun to be extracted from the input and compared with the commands I would allow at each location.

It was a simple, but impressive feat. Even more so considering that John wanted to fit the entirely of *Adventure* into the 15K program without losing any of the original game's features. Phil Garrett, reviewing the game in the second issue of Sinclair User speculated whether the lack of memory had forced some omissions, notably the particular "twisty maze of passageways". "Not a chance" he exclaimed, "on the first attempt I went straight into it like a homing pigeon. The maze seems like quicksand; the more you struggle to escape the deeper you sink."

[John] [My] version contained all the puzzles, locations and scoring system of the original version. The text was kept as close to the original as possible, but even with compression, some locations had to be abbreviated.

There was even space to include the original's famous magic verbs, and he worked in a very useful feature into his version of the classic treasure hunt. Typing BUILDING from anywhere in the game world [apart from the maze] would transport you back to the small brick building where you could deposit your ill-gotten gains. Sinclair User's Garrett was further impressed with its speed, programming, and ability to save your position to cassette. "It is an expensive program, but it is a remarkably good version of the original *Adventure* and well worth the money".

The included instructions began with the famous opening lines "Somewhere nearby is a Colossal Cave" and the game became notable in adventure history as one of the first commercial home micro version of *Adventure* to be available in Britain. Michael Evis [as Syrtis Software] ported a competent version to NASCOM in 1981, Level 9 weren't far behind with *Colossal Adventure,* and Artic Computing had pioneered a year previous with several titles, albeit on a different theme.

The more powerful ZX Spectrum was launched during development of the original game, and the extra 16K of RAM meant the abrupt location text required to squeeze the game into the confines of a RAM pack could be expanded. In comparison, John's new version included all 140 locations, a similar dictionary and all 64 objects from the original adventure, though it did exclude a knife-throwing dwarf and a pirate to guard the treasure chest.

[John] The Spectrum was just really a big ZX81, so I put back the missing text from the ZX81 version and wrote a 40-column text output routine rather than the [standard] Spectrum 32 columns so I could get more text on screen at a time.

The 48K version of *Adventure I* was released on the Spectrum in June 1982 [an advertised Dragon 32 version never materialised] at the very reasonable price of £9.95. Calling the game *Adventure I* opened an interesting discussion around the post-fixed number, and John suggests that further games in the series may have be considered. On the other hand, it may also have been an awareness of avoiding any copyright disputes, even though the ownership and licencing of the mainframe version on home micros was a very grey area.

[John] There were definitely big plans to write new adventures following the release of *Adventure I* and I wrote quite a few scenarios for these, none of which has survived the passing years unfortunately. [I was] very aware of IP, hence the *Pac-Man* game was called *Mazeman*, but University source code was pretty much thought of as public domain, rightly or wrongly.

Being the first faithful *Adventure* port to market certainly made a name for John, and Abersoft became well regarded in the industry. From selling small numbers on the ZX81, the mass market appeal of the ZX Spectrum meant that weekly sales were soon in the hundreds, instead of a few copies per week.

[John] It was a bit difficult to judge. I had some nice letters from people praising the game, one from a famous singer of the time who, jokingly, complained that it took ages to break a set as the roadies were all playing *Adventure I*!

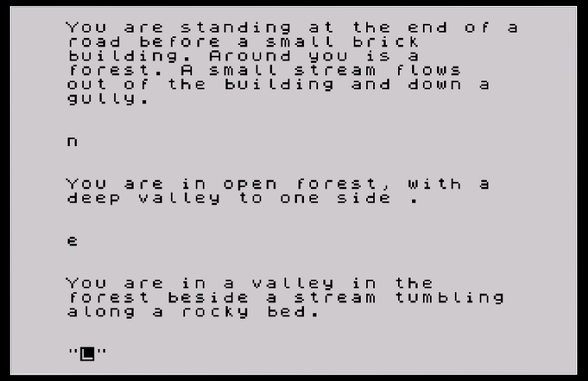
Adventure I was re-released twice; once by Buckinghamshire based CP Software as *Colossal Caves* and the version that most gamers recall, *Classic Adventure* [complete with Steinar Lund cover artwork] from Melbourne House – one of the emerging text adventure publishers in the British market.

[John] I had done some work for CP software, I forget what, and they wanted to have another game on their books so licensed *Adventure I*.

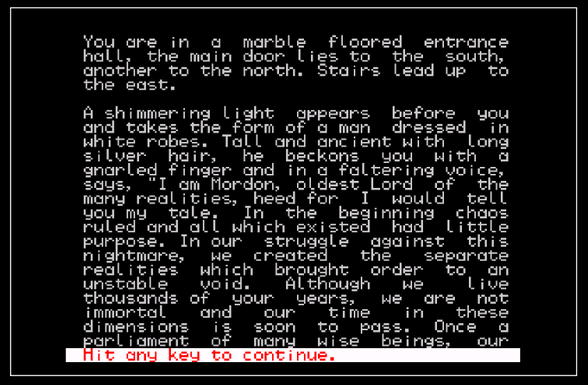
The re-releases gave John a chance to fix a bug that had plagued adventurers in the original issue. Having collected the bear, heading back to the bridge to release it to scare away the troll was met by a message that you weren't carrying the bear. A full disassembly of the game by John Elliot in 2016 showed the code replaced the "gentle bear" object in the game world with a new one, "bear following you", only to have the logic test for the original object later in the game.

The Melbourne House deal meant strong sales and an effective publicity campaign, lifting the marketing of the game from the humble classifieds into full-colour advertising alongside adventuring stablemates *Sherlock*, *Mugsy* and The *Hobbit*.

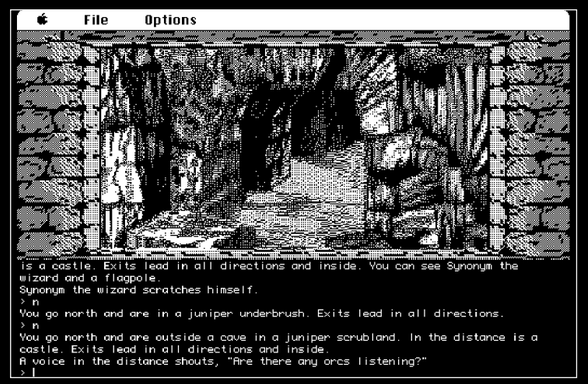
[John] Fred Milgrom called me one day and asked me to loosely join Melbourne House. It was more a case of being freelance with Melbourne House taking all my product. [...] We just found it quite strange to have a phone call [...] to the house and I don't know where they found the number from. But yeah, they just rang up and said, we want to market the adventure in Australia to start with. I did have a deal with a small company out in Australia, and I basically went back to them and said that I really do need to have this with Melbourne House because it looks like a strong link for the future. I gave them an offer to buy back the Australian rights, which Fred



[Adventure I] One of the first commercially available versions of *Colossal Caves* published in Britain. Abersoft's *Adventure I* is *the* classic treasure hunt, beautifully and elegantly programmed to fit into the confines of the 16K ZX81. The locations may be abbreviated, but all of the puzzles of the original are included. A technical marvel.



[Mordon's Quest] Jones-Steele partnered with authors Peter Moreland and Peter Donne for *Mordon's Quest*. He iterated his adventure system creating a lavish and complex adventure where the quest is to save the universe from destruction. No pressure.



[Knight Orc] John was employed by the Austin brothers to port Level 9 Computing's *Knight Orc Adventure System* to the powerful, but technically challenging black and white Apple Macintosh computer.

Milgrom paid so that he had no problems with copyright.

John was hesitant to give away his technology, but a deal with Melbourne House was too tempting and he had established a good relationship with their husband-and-wife management team. He had also grown tired of the administrative tasks associated with running a business, such as producing copy for adverts and duplicating cassettes. Now that Abersoft was achieving some commercial success, the time needed to fulfil those tasks started to overshadow the creative process.

[John] I got on well with [Alfred] and Naomi his wife and never had any real issues. [...] [Running Abersoft] I didn't like handling advertising, and I didn't like all the handling of the production. I wanted to program; I didn't want to deal with anything else.

After the monumental success of *The Hobbit*, Melbourne House was perhaps the best-known publisher of adventures in Europe, and becoming a part of that organisation going forward was a lucrative prospect. They promised to throw their growing might behind the game and funded John to port his source to other machines.

Classic Adventure was ported to the most popular home computer formats with John handling all the conversions himself: The BBC Micro, Acorn Electron, Commodore 64, Commodore C16 and MSX all received their own versions. Melbourne House's commercial weight resulted in each version receiving plenty of press and reviews in their appropriate dedicated press. Commodore Magazine said it was "one the best versions of that adventure for the Commodore 64." To demonstrate its continental appeal, the Netherlands publication MSX Computer Magazine said "*Classic Adventure* offers enough suspense, especially for the beginning adventurer, and is a good introduction to playing adventures."

[John] They all sold reasonably well, [and][...] I did all the conversions. I was quite jealous of [giving away] my system as it had such a small footprint I didn't want to let others have the ideas.

For the Amstrad version John was approached directly by the manufacturer's own software arm, Amsoft. Owner Alan Sugar knew that the successful launch of a new computer into a crowded market relied on the availability of software, and he saw games as a key component for his strategy at making the machine a success. Amsoft had one goal: obtain software to be bundled with the machine or get it onto the shelves for its launch.

[John] Amstrad got in touch with me when they were developing the CPC and I had one of the early computers along with along with a monitor that they said don't put your hands too close to it because it's not very safe. It had quite good ROM routines for handling things, rather than having to write my own. It was it was almost a straight port from the Spectrum just putting in the input and output routines.

Working freelance for Melbourne House suited John, but now there was a new offer on the table from Alfred to become a full-time employee. He also had an offer of a promotion into a management post for the Computer Department at the University. He weighed up the options, and remained with the University, renewing his publishing deal with Milgrom and continued to dedicate time to rewrite his adventure game engine.

[John] By the time I'd finished *Adventure I* and it was on the market I could see better ways of doing things. [...] In the evenings at the University I was writing out scenarios for different adventures that I would probably do in the future [and adding those logic features into the new engine.]

Whilst manning the Melbourne House stand at a micro show, John was approached by two budding adventure authors. Peter Donne was accompanied by Peter Moreland, a Production Manager who was already working in the games industry with Beyond Software. They both had an idea for a compelling text adventure narrative.

[John] Peter Moreland came up and said, we've got an idea for an adventure game, but we don't know how to write it. Would you write it? [...] I don't know whether he'd seen anybody else at that time, but he said we like your game and we'd like to have the same engine basically behind it. [...] We decided to work together on the title. They handled the design, layout and text and I developed a simple system for them to be able to write the game on paper and then input it into [...] version 2 of my adventure system.

The next iteration of John's adventure system saw him take tentative steps into producing a fully reusable ecosystem. Whereas the machine code of the original game did have the traditional separation of interpreter from the game's data, it was never developed into a format that could be easily adopted for new narratives. The new tool became a hybrid, akin to an assembler that provided a way to load high-level commands for compilation into faster, machine-code level instructions. It allowed non-technical designers such as the two Peters to efficiently describe the required adventure logic.

[John] The new engine gave them things that they could actually write out on paper and pass it back to me. I'd then put the codes in. So, there was different codes for jumping to a new location, or for parsing a word and they could write the whole adventure from that. All I did was take their notes and shove it straight into the computer. [...] They'd put down things like NORTH and then say what the description was and what you find there and what you do. [...] I'd basically [input] that in a fairly simple way and output byte code that the interpreter could handle.

John's creativity was at the periphery of the story and puzzles, making changes to the engine and the parser as necessary to accommodate Moreland and Donne's writing. Sinclair Programs in 1985 reported that the game was passed to John in book form, but this wasn't the case.

[John] No, it was it was being written as it went along. [...] I got the first area with the first few locations just to prove that the system would work. And so, yeah, so they [worked on] it following my notes. They wrote their script, sent it off to me. I programmed it in and sent it back to them for them to have a have a look at and then it went on from there until it was completed.

It was a slow process, with hand written notes and cassettes of updated versions of the adventure going back and forth between the authors and programmer using the postal service. Despite his new employer insisting that their adventure catalogue contained graphics, the second version of the *Adventure I* system didn't employ any imagery at all.

[John] At the time, I thought that the text was more important than graphics and that graphics just stole my memory! Once the new generation of computers came out, I could see the sense in having both great prose and graphics that looked like something.

Mordon's Quest [taken from the first three characters from each of the writer's surnames] was released in July 1985 for the ZX Spectrum, Amstrad, Atari and Commodore 64. A BBC version, listed on many internet repositories was never produced, though John speculates that a version would have been viable given that the Atari version ran on the lowly 600XL. In 1986 *Mordon's* was packaged with *Adventure I* [renamed simply *The Adventure*] in a classy two-in-one collection called *QL-Classic Adventures* for the Sinclair QL. The games were shipped on Sinclair's notoriously fickle magnetic mass-storage Microdrives especially for the high-end computer.

[John] I seem to remember a meeting at Sir Clive's house. I would imagine it was a straight port, maybe with more than 40 characters per line. I believe the rights to the adventures were negotiated for each platform and Melbourne House weren't interested in the Sinclair QL.

Mordon's Quest had the player travelling across the galaxy in a time machine, collecting lost pieces of Mordon's immortality device to save the universe. John and the two Peters took full advantage of the extra memory gifted by excluding graphics and crowbarred 150 locations and an expansive 400-word vocabulary into the game. There was a chance to be more expressive with the prose, and it looked very impressive on screen thanks to a new text handling system that widened the usual Spectrum display to create the impression of a full page of text.

Adventure critics welcomed what they saw as the return to large-scale text-only games, preferring the "flavoursome" descriptions, "evocative prose" and wealth of text that bucked the trend of concise descriptions found in graphic adventures. Your Computer magazine said "with its detailed atmospheric text and touches of humour, *Mordon's Quest* guarantees many hours of stimulating pleasure". The Spectrum flagship magazine Crash raved about it, saying "it has long, atmospheric puzzles ingeniously interwoven into a fascinating fabric of clues and apparent dead ends." At the Crash Readers Awards in 1985, they awarded it the winner of the Best Text-Only Adventure game of the year.

One of *Mordon's* biggest advocates was venerated Computer & Videogames adventure journalist Keith Campbell. Keith loved the game and was impressed with the enormous amount of text that John had managed to cram in using the latest version of his text compression routines. He waxed lyrical saying *Quest* contained "those deliciously funny and exciting surprises that make an adventure a pleasure to play" and that it was "exciting and humorous [and] always interesting and never predictable." Keith gave it the maximum score possible, 10/10.

Based upon that praise, *Quest* should have sold exceptionally well, but confusingly Melbourne House advertised *Mordon* as a continuation of *Classic Adventure*. Since it drew no narrative parallels with the original game it was misleading for adventure fans, prompting them to believe it was more of a game for traditionalists other than something that offered a richer experience.

As for further games, the opening text spoke of being the "first in the saga" and the ending contained a tantalising indeterminate THE END? to suggest follow-up games in the series. Keith Campbell at the foot of his glowing review reported of an imminent sequel called *Bostafer's Revenge* – it never materialised.

[John] Apart from some initial discussions nothing happened, [and] nothing more was done. Peter Moreland got a job in the industry, and the time for a new game just wasn't there. [By the way] Bostafer was an anagram of Abersoft if you hadn't realised.

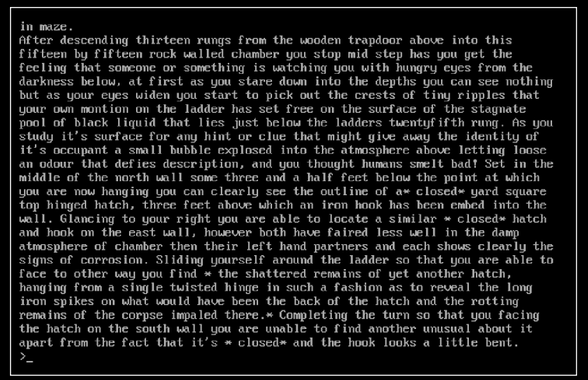
In late 1985, Peter Moreland moved into a position with Telecomsoft when Beyond Software was absorbed into telecoms giant after being sold for just under £1,000,000. Though it dashed any chance of further text adventures in the *Mordon* series, Peter's move would have repercussions for John's later career.

The part-time agreement with Milgrom had also started to effect John's perception on being able to produce games in-line with an ever-changing delivery schedule. Melbourne House seemed to want to bring more staff in-house, and in the end a minor dispute over royalty payments strained the relationship resulting in John terminating the agreement.

[John] My deal with Melbourne House didn't really give me a reasonable percentage of the sales and that was the beginning of the end for the relationship. I spoke to [Level 9 Computing] at one of the [micro] shows and at the time they were looking to expand. They wanted to put their games onto the Macintosh, and they were finding it quite difficult to get the graphics to do anything sensible on black and white monitors. [...] Pete and Mike [Austin] made me an offer I couldn't refuse to leave the University and work for them full-time. I still worked most of the time from my office at home but occasionally spent a day or two down with them discussing the projects.

The initial period of work for Level 9 was positive. John spent his time porting their existing games catalogue to the Mac and then assisted in converting and optimising parts of the Austin's adventure system to different machines. In terms of how he was managed it was a hands-off situation.

[John] Then I was beginning to find it quite difficult [...] working with a family firm. Everything was fine in the development, but ideas I



[Above] The final IBM PC build of Paul Copping's and John Jones-Steele's *Rainbird Adventure Implementation Language*, playing the example test adventure.

---

had seemed to be put aside. It wouldn't have been a relationship that would last long. I had been in charge of my own department for years at the University and it was difficult for me to have to write things I wasn't convinced were correct.

John is credited for his contribution to the *Knight Orc Adventure System*, or *KAOS*, for Level 9, an expansion on their *A-Code* technology. The engine was used for *Knight Orc*, and subsequent *Gnome Ranger* games, the Mandarin published *Lancelot* and Level 9's final game, *Scapeghost*.

Looking for a way out, John's tenure at Level 9 ended in 1987, shortly before the release of *Knight Orc*. He was approached by Telecomsoft to write a new adventure system for Rainbird. Though he can't recall, it's likely that John was recommended for the position or headhunted by former adventure collaborator, Peter Moreland.

[John] The plan was to have an adventure language system that they could then employ authors to come up with an adventure, and it could be easily programmed by anybody rather than most of the adventure systems where things had to be tweaked as they went along. If they didn't have what you wanted in there, the program would have to add it in. The idea was that we're trying to make a language that was all things to all people.

The *Rainbird Adventure Implementation Language* [or *RAIL* for short] came from the small, experimental footsteps taken with *Mordon's Quest*, evolving from the simple logic editor, into a larger, commercial-grade system that could be reusable for multiple game scenarios. Each adventure would be designed using an abstract, high-level C-like language that would be compiled on multiple platforms using a single source code.

[John] *RAIL* was a work in progress, Paul Coppins at Telecomsoft was testing the system and getting me to add new features he wanted. Paul could write in *RAIL* and run the compiler. It would compile it for any system, so you'd only ever have to write it once rather than having to change it [over and over again]. That was the theory.

Nicknamed The Axeman, Coppins was the resident adventure enthusiast at Rainbird, having previously been a voluntary font of knowledge for Keith Campbell's Valley Rescue and Adventure Helpline. He joined Rainbird in December 1986 as an adventure tester and set to work crafting an epic narrative called the *Light of the Shadowlords* that he used as a testbed to give *RAIL* a real-world grounding in its evolution.

[John] It got quite good at the at the end, because Paul was writing stuff and every time he came along said, well, I want to do this, but I don't know how to do it yet in the engine, I'd write something new and then he could make that next bit work. So, it was very much an iterative process in there. The idea was that he would keep trying to put things in that he thought every adventure writer would want and I'd work out a way of making it so that anybody could write it.

The system was in development from 21st November 1987 until 21st June 1988 with John and Paul working closely to produce an engine that had advanced object handling, industry-leading text compression, and a sophisticated parser that was set to challenge the best that Infocom, Level 9 and Magnetic Scrolls had produced. As well as the planned in-house releases from Rainbird, the tool would be commercially viable and potentially could be sold to external developers to create their own adventures – at a cost.

[John] The idea, I believe [was] to package it and let other adventures writers publish things with a royalty to Telecomsoft for the use of the engine. [...] I think it was probably too late. if we'd started on it at the time I joined Level 9, it would have probably been mature enough by the time Telecomsoft wanted it. [...] There was a lot of wasted time in writing games the old way so it would have been nice to get it finished.

Unfortunately, the promise shown by *RAIL* was never realised, it was shelved soon after the Summer of 1988 as the commercial text adventure market dwindled with the advent of point-and-click games and consoles. Richard Hewison, on his defunct Bird Sanctuary website reminisced about the end of the product: "One of the main issues with *RAIL* was that it was an adventure programming language rather than a menu driven system like Incentive's *STAC* [on the Atari ST]. Even though it would have produced technically advanced adventure games, it might have proved difficult to use for a beginner. That main issue, coupled with the diminishing commercial viability of text entry adventure games probably contributed most to the premature end of the project."

[John] After Telecomsoft was bought by Microprose, I went back to running Abersoft before joining Goliath Games as a partner. After Goliath closed, I was back to Abersoft where the only adventures I had a hand in was the *Ultima VI* conversion for Amiga and ST. There was some initial work done on *Martian Dreams* [part of the *Worlds of Ultima* series] but that didn't happen. Trying to find publishers for adventures, especially text adventures, was impossible so I had to turn my hand to anything that was available.

John has spent a lifetime career making games, and approaching 70 is still coding for the latest Sony and Nintendo platforms. He speculates that the end of the text adventure genre was down to the age demographic of players lowering as consoles started to dominate the market. As for the continuing debate around the fragmentation of British text development companies and their unwillingness to share and consolidate technology, he proposes that Telecomsoft should have acquired Level 9 and Magnetic Scrolls.

[John] [At PCW Shows] there was always the adventure evening out where everybody in the adventure industry went out for a meal and drinks, sitting around a big table and there was always good fun and it was very friendly. [...] It was very much a case of we're doing our thing and you're doing your thing and never the twain shall meet. [...] It would have been better for Telecomsoft to do what modern companies do if they find something they like and buy the company and get the whole lot. But in those days, it was a case of we quite like this, but we're not getting on with the management of the company. So rather than buy them out we'll reinvent the wheel and do the same thing they're doing, just with our own people.

Some of John's contributions to Level 9's *A-Code* system can now be explored through Mike Austin's preservation project available on GitHub [see **Level 9**]. As for the Abersoft source, it is sadly lost to time.

[John] No, it's very much the case that all of the backups that I did on both the Amiga and Atari ST got corrupted at some stage. [...] It's all gone, so that's a pity we didn't have GitHub back in the day.

DESERT ISLAND DUNGEONS

Run aground on an errant piece of coral and sinking fast, **John Jones-Steele** is forced to abandon the good ship Abersoft and becomes a castaway on Dungeon Island with only five text adventures to play until help arrives.

***The Zork Trilogy*, Infocom.**
The spiritual successor to *Colossal Caves*. I could get lost in this for ages.

***The Hitchhiker's Guide to the Galaxy*, Infocom.**
Brilliantly brought to life the book!

***Leather Goddesses of Phobos*, Infocom.**
Just really silly but very entertaining.

***Trinity*, Infocom.**
Thought provoking and probably the best text adventure written.

***Planetfall*, Infocom.**
Great Sci-fi adventure. Floyd the robot was a brilliant part to the game.

TWILIGHT INVENTORY

Gareth Pitchford knows a thing or two about what makes a good adventure. In **Twilight Inventory** the author of *Microfair Madness* and *Get me to the Church on Time!* takes us on a whimsical trip back in time to explore the flourishing British indie adventure scene of the 1990s.

Author: Gareth Pitchford
Publisher: Self-published
RRP: £9.99
Buy it from: Lulu, Amazon
Website: http://www.8bitag.com/

Twilight Adventure is a love letter a better time. A time when an entire genre of games was supported by an underground collection of indie developers, fanzine writers and secret guild aficionados. Gareth Pitchford's book aims to "give the reader a flavour of the small, but vibrant and flourishing British adventure scene" and he does it with aplomb. Being a former adventure author himself, with several games under his belt [the majority published on the funky Delbert The Hamster Software label] he's well placed to know what makes a good puzzle.

He's done a terrific job with the book, it's a lovely volume, and available via the self-publishing site Lulu, Amazon [physical or digital] or as a free [yes, free] PDF download via the author's own website. He's pulled together a plethora of reviews [written back in the day for a variety of adventure 'zines] into a 234 page paperback that covers a varied range of games and authors, many of which you won't have encountered before. It features publishers from The Guild, Compass Software, GI Games and FSF Adventures through to Zenobi. Gareth has left no stone unturned.

It's well presented, with the cover featuring a delightful piece of his own trademark pen and ink artwork, and it's designed to look like a cassette inlay – though the effect is only really apparent with the digital version. Each game is given a four or five page review, with a couple of screenshots for each [all in black and white in the print version] game. Gareth's writing style is witty, relaxed and his extensive knowledge of the games shines though. It's witty, charming and puzzles and characters come to life with a wry sense of humour. He's able to bring a different viewpoint to the game's mechanics, often from the advantageous position of knowing the authoring software [usually *The Quill* or *PAWS*] techniques inside out, and explores little explored nooks and crannies in each of the titles.

As suggested earlier, it's a collection of old reviews, whipped into shape for the book, and the tense still refers to the games "of the time". There's lots of recommendations to rush out and buys, and text looking forward to forthcoming games, but for those Gareth has added a small postscript in italics to bring the text up-to-date. It's a nice touch. In the back there's a useful index, a summary of publishers and authors, and a brief look at some of the fanzines around at the time. He's even included a few SAM Coupé games for good measure, though who had a Coupé, right?

It gives old-school text adventure aficionados a reason to keep those rose tinted glasses on for a little longer. At under a tenner, Twilight Inventory is an absolute steal. For free it's a no brainer, just remember to make a small contribution to Gareth's charitable cause.

EXPLORING COLOSSAL CAVES

Somewhere nearby is **Colossal Cave**, where others have found fortunes in treasure and gold, though it is rumoured that some who enter are never seen again. Magic is said to work in the cave. I will be your eyes and hands. Direct me with commands of 1 or 2 words.

Bedquilt Cave Adventure – do you remember that one? No? Well, in a parallel world it may have been the title of the text adventure that started it all. Bedquilt was part of the Mammoth Cave system in Kentucky that gave Willie Crowther the inspiration for the original *Adventure* game. He told Dennis Jerz that "the geometry [of the game] was lifted directly from Bedquilt Cave" not from Colossal Cave.

Crowther had started writing *Adventure* – an interactive textual simulation of a caving expedition in 1975, as he was going through a protracted divorce. He used the development of the game as a way to engage with his children in his spare time. Along the way, the game evolved to include puzzles and mythical creatures, more than likely under the influence of the Dungeons and Dragons games he was playing with a group of friends at the time.

His first effort, 79 locations and 193 words in 700 lines of FORTRAN code and 700 lines of DATA was saved as ADVENT, limited to six characters because of the restrictions placed on file names on the PDP-10 mainframe computer he was using. In fact, though the game is often referred to as *Advent*, there's no doubting its name - the text always began with the message "Welcome to *Adventure*".

Crowther left for California to further his career in 1977 leaving *Adventure* behind. He had no ambition for the game other than to keep his children occupied and to share it with the computer community. It was rumoured that once he'd overcome the challenge of creating the program he lost interest in it.

Luckily for us, others did maintain an interest and the unpolished code made its way onto an experimental computer network at the US Defence Department's Advance Research Projects Agency [ARPAnet] where it was discovered at Stanford AI Labs by computer science

student Don Woods. He was instantly hooked, telling Matthew Lyon in his book Where Wizards Stay Up Late, "*Adventure* made users feel like they were interacting more with the computer. It seemed to be responding more to what you typed, [and] I think that attracted a lot of players. [...] This was playing with a computer."

Woods attempted to track down the original author by doing the only thing open to him in pre-internet times - he sent an e-mail [another first created on the ARPAnet system] addressed "Crowther" to every other network that was in existence at the time. Amazingly he got an answer, and received the source code with Crowther's blessing on the sole condition that he would return a copy containing any changes that were made. Woods quickly got to grips with the source, taking about four months to fathom out the code, fixed some of the original's bugs and corrected problems with several location connections. That became version 1 and became immensely popular, quickly finding its way to almost every computer network in the US.

Version 2 took around a year, doubling the rooms, adding new treasures, puzzles, expanding the text, and introducing a pirate who stole the player's treasure. Woods told Jason Scott for his GET LAMP documentary "I tried to make the various puzzles within the game interlock, [...] this thing that happens here matters later, and you thought you were done with this part of the cave but there's something else you can do there."

A more Tolkien-esque feel to the world was introduced, and Woods told Eric Raymond, responsible for creating *Open Adventure* [a recent port of version 2.5 from 1995] that the main differences were the introduction of "new treasures" that "require[d] solving a puzzle that's definitely at the tricky end of the scale for *Adventure*." The forest was converted from a partially-random maze with a few locations to a fully-fledged maze, the scoring system was tweaked, and the game's troublesome torch and finite battery life was overhauled.

Version 2 was significant because it also introduced the concept of basic Artificial Intelligence and Pseudo-Intelligent Characters to an adventure. Woods changed the simple behaviour of Crowther's Dwarves. They had been programmed with a fixed route of movement, meaning that in the new expanded world it would be very unlikely that the player would ever encounter them. Woods introduced a "wander" algorithm, giving the characters the ability to move around the cave at will, but adhering to the rules coded into the database's location connection tables. He also added simple AI, for example, ensuring that all of the Dwarves didn't end up stuck in dead-end parts of the caves. "I had to make them a little smarter" he told Scott.

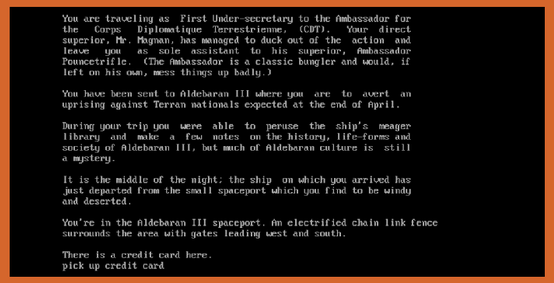
Between 1977 and 1995 Crowther and Woods continued to work intermittently on the game, culminating in the release of Adventure version 2.5. To say we owe them both a great of debt of gratitude is an understatement. *Adventure* is as important videogame history as *Spacewar*, *Space Invaders* or *Pac-Man*. It defined the conventions that we take for granted, and it's DNA can be traced in every single text adventure that has come since.

We can also be thankful that in those pioneering days, there was a culture of experimentation, and that software was social and something to be shared. Both Crowther and Woods wanted the game to be communal, modified and enjoyed - though neither expected it to be something that people paid for. Woods is thankful for the occasional donations he receives, every now and again, and says that thanks received in an e-mail never gets tiresome. On the subject of royalties for *Adventure*, in GET LAMP he shrugged his shoulders and told Scott "I didn't mind if people wanted to take a copy and play with it, and make a copy available for free – when someone took it and turned it into a version they charged for it and didn't give me anything for it – that was annoying – but what can you do?"

> "I didn't mind if people wanted to take a copy and play with it, and make a copy available for free"

A WANDERING STAR

We rightly celebrate *Adventure's* pioneering place in history, but it is argueable that Peter Langston was the first with *Wander*, a text-based world modelling program he coded in 1973 whilst teaching at Evergreen State College in Washington.



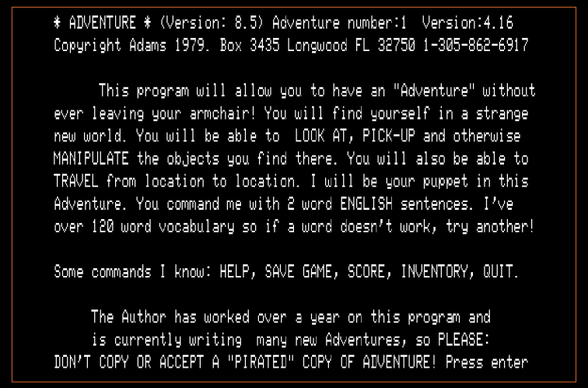
Written in a mainframe version of BASIC, it wasn't a game as such, more a framework for the construction of games. *Wander* enabled the creation of fantasy stories containing rooms, states and portable objects that could be affected by decisions made by the player. The original description refers to itself as "a tool for writing non-deterministic fantasy stories" - but it was much more than that, including some of the verb/noun and directional standards that would later be attributed to *Adventure*.

Langston created several demonstration games with *Wander*, including *Castle* where you explore a rural area and a castle searching for a beautiful damsel. He wrote other, more successful games such as *Empire,* and later went on to work for LucasArts.

A copy of *Wander* has recently been discovered buried in a software distribution from the Usenix 1980 conference. Enthusiasts are currently working on Windows, Mac and Linux ports of the source code.

COLOSSAL CLONES

Adventure has appeared across virtually every computer format in abridged, full or expanded guises. **Classic Adventurer** starts its journey standing on the end of a road, and peers into a small brick building that houses a few notable examples.



ADVENTURELAND

Scott Adams, 1978

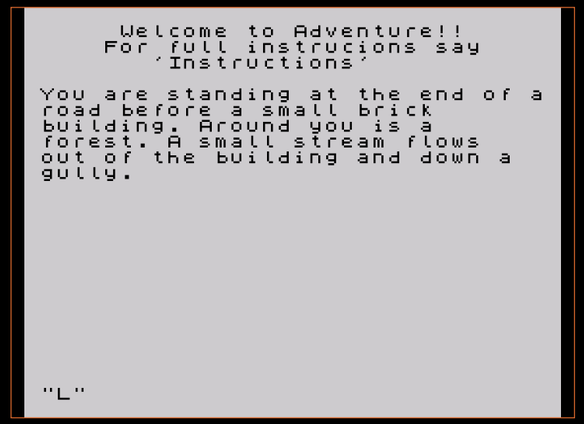
Inspired by *Adventure*, Scott Adams created his own minimalist version, *Adventureland* for the 16K Radio Shack TRS-80 computer in 1978. He had the idea to produce an adventure interpreter, a rudimentary "adventure engine" that would enable him to easily create other games. *Adventureland* was soon converted to a wide range of machines, and can be credited as the first mainstream commercial text adventure available on home computers. For many it was their first exposure to the genre.

MICROSOFT ADVENTURE

Microsoft, 1979

Written by Gordon Letwin, *Microsoft Adventure* was among the first four products sold by Microsoft in a new division created to sell software to consumers.

The game is completely faithful (albeit with an extra 'software den' room in a nod to Letwin's full-time job) to Crowther and Woods' program by cleverly accessing the original's huge databases via a required floppy disk drive. It was one of the first games to be retailed professionally and Microsoft released a version of its *Adventure* in 1981 with the initial version of *MS-DOS*.



ADVENTURE I

Abersoft, 1981

One of the earliest commercial text adventures to appear on British home computers was Abersoft's *Adventure I*. It was written by John Jones-Steele for the Sinclair ZX81 and packed a faithful [and fast] reproduction of *Adventure* into 13K, including the majority of the original's locations and puzzles, albeit with shorter descriptions and a lesser points tally required to complete the game.

Adventure I was expanded by 32K into a new version for the Sinclair ZX Spectrum machine. It utilised the same separate logic engine and database with a clever proportional text routine that vaunted the expansive prose of the extended game.

The game was re-released by CP Software as *Colossal Caves* and in 1983 Melbourne House re-packaged and ported the game onto a wide range of micros giving it much wider mainstream prominence as *Classic Adventure*.

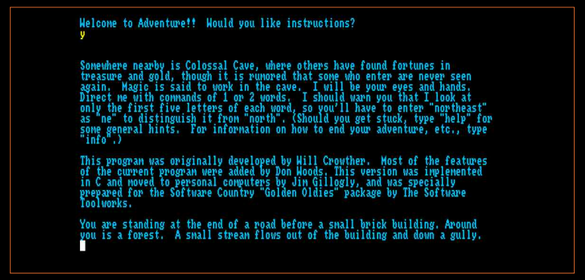
THE VERY BIG CAVE ADVENTURE

CRL/St. Brides, 1986

Not even the great Crowther and Woods could escape lampoonery, and *Adventure* fell foul to a terrific parody from the ladies of St. Brides in the form of *The Very Big Cave Adventure* in 1986.

Trixie Trinian guides us through a game full of puns, jokes, and comic put-downs in this cracking spoof. There's lots of familiar locations, most with a twist though, for example the famous well house has a spring, but not of the watery kind, and several other surprises await, such as the inclusion of a *Space Invaders* room where you can play the game in text form.

It has the usual *Quill'd* polish from a CRL title and even includes sounds, visual effects and a RAMSAVE feature.



THE ORIGINAL ADVENTURE

Software Toolworks, 1981

Software Toolworks sold a modified version of *Adventure* in 1981 as *The Original Adventure*. It was endorsed as being "in the spirit of the original game" by Crowther and Woods, in return for a modest royalty, and remains the only version of the game to pay the duo.

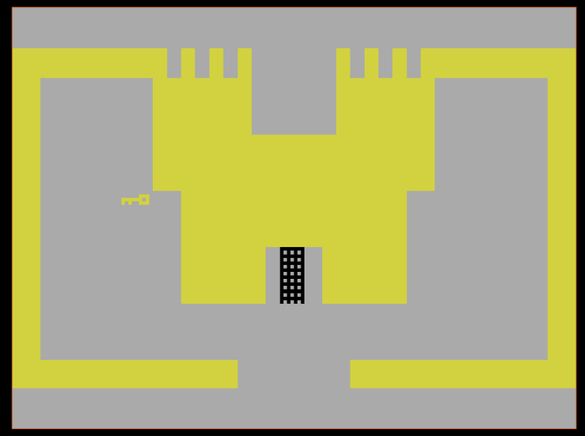
Any player who earned all the points and found all the treasures were shown a secret that could be sent to the publisher in return of a Certificate of Wizardness signed by the authors.

COLOSSAL CAVE ADVENTURE

Duckworth, 1985

Jim Butterfield created one of the first *Adventure* clones for the Commodore range of computers with his version of *Adventure* for the PET in the late 1970s.

Butterfield's version required a disk drive to store and access data, so would be unsuitable for the British cassette-based market. This version was coded by adventure royalty Peter Gerrard. He created a port called *Colossal Cave Adventure* for the more accessible Commodore 64 and Amstrad computers in 1985.



ADVENTURE

Atari, 1979

Okay, okay, so we may just be stretching the limits of imagination to include Warren Robinett's 1979 version of *Adventure* as a true text adventure. But, it's ambition alone is worthy of discussion, and displays a feat of programming that stands alongside Scott Adams and the Austins of Level 9 for its sheer genius.

Robinett was inspired by a visit to Standford Artificial Intelligence lab where he spent many hours playing *Colossal Cave Adventure*. It was June 1978, and he worked for Atari, where he was charged to write a new game for the Atari 2600 home videogame console.

His design had the player moving around with the joystick, showing one room at a time on screen and showing each object in the room as small graphics. It was the game's major innovation - the idea of moving through a network of screens that matched some of *Adventure's* locations meant that the action of the game could take place in much larger and more interesting space than the single-screen experience of most videogames of the time.

He reduced Adventure down to around 30 rooms, and begin making each of the characters and objects into recognisable objects. The characters moved, all with a basic form of artifical intelligence, and all crammed into 4K of cartridge ROM. Robinett told Wired.com in an interview "I even had 15 bytes of RAM left over. There was room to do three more dragons if I had chosen to do so, but it seemed to be working pretty well. I guess that's what you'd call game balancing nowadays."

It was hugely ambitious and complex at the time, blurring the lines of adventures, role-playing and rogue-alikes into one game. It even had one of the first examples of a videogame "Easter Egg" - a secret room that had the author's name within in.

COLOSSAL ADVENTURE

Level 9, 1982

Whilst working at a mainframe manufacturer and distributor Peter Austin discovered *Adventure* during one of his lunch breaks. He spent hours solving the game and set his sights on something pretty ambitious - a full conversion to a NASCOM machine with a paltry 32K of available RAM.

Working with his brothers, they formed Level 9 and devised a re-usable adventure system in a language they dubbed A-Code. It was so ultra-efficient at compressing text they were able to expand the original game's location count to 200, add extra puzzles and implement a cunning new end-game sequence. They called their new version *Colossal Adventure*.

Because A-Code was machine independent, it meant that *Colossal Adventure* was ported to almost every other home micro at the time. The Nascom, BBC, Spectrum, Commodore 64, Oric, Atari, Lynx, MSX and Enterprise all received a version.

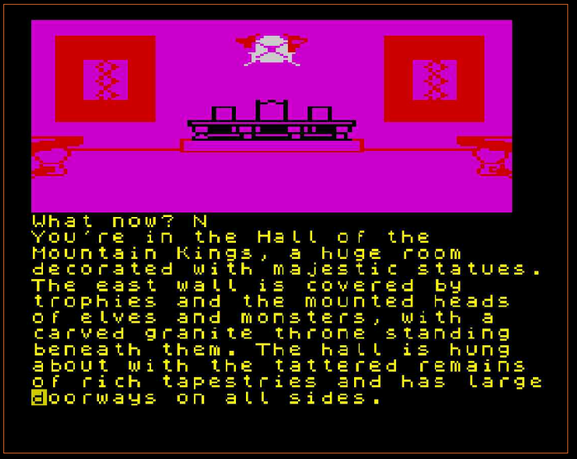
If you didn't own Melbourne House's *Classic Adventure*, then you'd own Level 9's *Colossal Adventure*.

It was later re-released as part of the *Jewels of Darkness* trilogy alongside it's sequels *Adventure Quest* and *Dungeon Adventure*. This 1986 version was significancy revised to incorporate the latest version of A-Code which allowed for the inclusion of location graphics alongside the descriptive text.

DELVING DEEPER

More caves to explore ...

Colossal Caves, Spectrum, Anubis, 1985
Adventure, BBC/Electron, Micro Power, 1983
Adventure, Apple II, Frontier Computing, 1981
Zork, PC, Infocom,
Acheton, BBC/Electron, Acornsoft
The Serf's Tale, Spectrum, Players, 1986
Analog Adventure, Atari, Analog, 1981
Pyramid 2000, TRS-80, Radio Shack, 1979
Adventure, Kaypro II, Quest, 1983
Adventure, ZX-81, Bug-Byte 1982



LEVEL 9

Long regarded as Britain's answer to Infocom, Level 9 Computing produced adventures that outpaced their American counterparts in speed, sophistication, and sheer ingenuity.

At the start of the 1980s, brothers Mike, Peter and Nick Austin were writing arcade games and utilities for the Nascom computer in their spare time and marketing them in the small ads of the emerging home computer press. They called themselves Level 9 Computing.

[Mike Austin] We formed Level 9 in September 1981 and the reason I could be definitive about that was that my parents weren't happy for me to run a business until I was 16 and I turned 16 at the end of July 1981. We gathered some pennies together and formed the partnership in August ready to launch. [...] I was a big fan of early computer games – both in arcades and on the first home computers. The top level on a lot of these games was "level 9", and in any case, you can't get any higher without using more digits, so we went with that.

To earn a respectable living, Pete Austin started his working life as a programming consultant and endured 18 months of rigour writing enquiry packages for financial institutions. He moved onto receive an invaluable grounding in computer architecture and hardware by writing system software for mini-mainframe manufacturer Perkin-Elmer. He drafted in his brothers to run a side-line business combining their computing passion and experience to create a series of utilities.

[Mike] I produced a product called *Extension BASIC* which was for *Microsoft BASIC* the Nascom and it added a load of structured programming concepts into the *BASIC* interpreter. [...] It was what we call these days call a code injection attack because the interpreter was never designed to be extended. But I found that there was an unsafe vector which I was able to leverage and effectively exploit and insert my own code into the execution path. [It allowed] for the interpreter to understand all these extra concepts, which was fun.

At Perkin-Elmer, Peter chanced upon a game stored on their network called *Adventure*. He was a huge fan of Dungeons and Dragons at

University (in a time when the game's rules were only available for purchase in a single shop in London) and *Adventure* fired his imagination to spend most lunchtimes immersing himself in Crowther and Woods' fantasy world. "There had been a number of simpler games on the computer at Cambridge, based on the D&D format. *Adventure* was full of puzzles, many of which were extremely unfair. I cracked it in two weeks" he told Micro Adventurer.

He was keen to be able to play a version of the game with friends and family, but without access to any machine capable of running such a piece of code, he decided to write his own with the help of his brothers. Where others had adapted or ported a more minimal adaptation of the game, the Austin's wanted a full version true to the original. It was a hugely ambitious undertaking, given that the Nascom had a meagre 16K of RAM.

[Mike] The Nascom was actually Nick's computer that he built a couple of years before that. I'd been teaching myself to program over hat couple of years. £40 was the initial capital [and] we had relatively modest aspirations to start with in terms of, you know, placing some adverts in the magazines at the time and seeing how they went. [...] **We didn't have access to [any of the original] source code at all, and to be honest it wouldn't have been much use anyway because it would have been far too big to run on the computers we were targeting. We started off just trying to re-create *Adventure* as I wanted to play it. We [worked through the game] solving the puzzles and I think we probably had access to a printout of the game text too. [...] Peter [...] to some extent he was relying on his memory as well. He wrote down what he thought had to happen, I then implemented it and then he'd say, "Oh well, that doesn't play quite right, so let's change that."**

For the challenge ahead, The Austin's needed to create an efficient ecosystem using the Nascom and its Z80 processor that would enable them to quickly build games. To implement the *Adventure* logic, they rejected the available methodologies that were around at the time – *Scott Adams' Adventure System* and Ken Rees' *Adventure II* thesis - and decided to draw upon their experience and expertise to create their own language.

Pete told Atari Users Magazine that "we didn't write a game, but we actually wrote a system which interpreted a database. Virtually, everybody else writing adventures at that time, in fact everybody that I can think of, went and wrote one-off games."

[Mike] I was fascinated by all types of computing, and I was looking at a lot of the early programming languages. [...] We did a version of *Forth* which was written by somebody else and we published it. But the conceptual idea for *A-Code* came from *Pascal*, which was compiled to *P-Code* [an early portable type of machine language which could be interpreted on different computers]. I did an early compiler called *Q1*.

Q1 was Mike's initial attempt at writing his own abstract, machine-independent language. Instead of compiling a source program

directly on the home computer it was designed for, *Q1* would be that intermediatory step – halfway between human-readable code and hardware-specific machine code. It could be written once, then interpreted and run on wide range of machine ... well, that was the theory.

[Mike] *Q1* was designed to be extremely quick to execute, be very compact, and portable across computers. We used that to create some early arcade-style games like *Nightmare Pork*, [and] various versions of *Space Invaders*-style games. [...] there's a whole bunch of them which had to have relatively fast execution. You couldn't do them in *BASIC*, but equally I wanted them to be portable because at the time there were so many different computers. Even in those early stages there was a lot of fragmentation, but I thought If I compiled the logic to something like *P-Code* and write an interpreter how fast does that work out? It turned out pretty fast enough for those arcade and action games, so that was *Q1* and that was probably about 1980.

As with many other early developers going pioneering work in the early years of the industry, the foresight to create a language and game interpreter that was portable to many other home computer formats, either existing or those that would arrive in the future with the minimum of fuss was truly visionary.

[Mike] Well, thank you. Nobody told me it wasn't possible, so I went off and did it. I was obsessed with the whole thing, but when I started to think about how to implement *Colossal Adventure*, the requirements of the language weren't the same, but I liked the pre-compilation of *P-Code*. So, I thought, OK, let's do another language.

Fast and portable, *Q1* was generic and could be used for all kinds of game. Mike wanted something more focused on providing the functionality required for text adventures. He tailored the language to suit and called it *Adventure-Code* or *A-Code* for short.

[Mike] When I came to look at creating the adventure games, we had two main objectives: To create games that would easily port onto lots of different home computer systems, without re-writing, [and] to create very compact code, so we could fit sophisticated game logic into a very small amount of memory. None of the languages around at the time fit these requirements, so I created *A-Code*.

A-Code source looked a little like *BASIC* and when compiled created extremely small object files. It was a blueprint designed from the start to work well for adventures, so it contained instructions to simplify user interactions and text output to screen. Any author of a game didn't have to worry about how to display things, parse English input or handle text compression.

[Mike] There was some derivation in there from *Q1* as I put in features that were going to work for adventure games. We needed a parser, so I wrote a parser and that became part of the runtime for the language. It wasn't strictly part of the language; it's a separate runtime of library functions that you could use. It handled things like N, SE, W, UP and the coordinate system. [It was] standard stuff because the idea was that we put all the standard stuff that had to be reasonable quick and very small into assembly code. We put higher-level constructs into *A-Code* as it [evolved].

The effort required to be compact and efficient was paramount for developers aiming to sell any type of software to the fledging British market. Infocom, the US giants blazing a trail across the pond had been able to bring complex and expansive games to market much earlier. They relied on a £40 price tag and had access to an enormous domestic market with an abundance of disk drives in many homes. Infocom didn't have to worry about text compression or fast and efficient code, and in many games the response time was sluggish as the engine whirled away retrieving data from disk. British developers had none of these luxuries and the limitations of cassette-based storage meant Mike's code had to be a perfect exercise in miniaturisation.

[Mike] The interpreter was small, and the runtime was small and the actual logic for an adventure game is small – we're talking a few kb in total for all of that. The bulk of everything was always the text, and then later the graphics started to nibble away at that. Particularly on those 16K computers the text was very hard to fit in, and although the compression we had was very good from the first version, there was still a limit.

Creating compact and efficient code was one element of the *A-Code* suite. Compression would be the key to cramming as much data into the small memory footprints of home micros. The oodles of descriptive text in a text adventure was the obvious area of data to target.

[Mike] We went through several iterations of text compression algorithm over the years, all proprietary. The first one was, I think, based on Huffman encoding plus some dictionary lookup for common words. As with the text compression, we compressed the object code to squeeze every conceivable jot of functionality out of every byte. So, we used similar techniques – variable length instructions, instructions which contained common data values, etc. The first version achieved really good compression results. The later versions achieved staggeringly good compression, and were also used to help with parsing user input.

Pete outlined the deep thinking that was going into the complexity of Level 9's algorithms that analysed text patterns. He told Sinclair User magazine in May 1985, "It doesn't always pick up what you'd expect it to. In the phrase 'in the room' the compressor might decide that it was more efficient to use a code for *n th* and *e r* rather than pick out 'in' and 'the'. That is not something which occurs to the human mind."

[Mike] Later we realised that, actually, a lot of the words you're going to be using as an input are words that the game also might need to be able to output – and often at quite a high frequency. So, you know, I don't know, 'scabbard' in *Lancelot*, it's quite a long word but you need to be able to recognise it as a word for parsing, but you also need to use it a lot in the text.

The Austin's version of *Adventure* was entitled *Colossal Adventure* and took about a year to create. Keith Campbell heaped praise on the game, saying it was "the nearest to the mainframe version I played." It was an extraordinary technical achievement, managing to squish 200K into running on such a limited home computer. In fact, *A-Code* and its text compression routines were so efficient, on the 32K BBC Micro it yielded an unexpected and incredible bonus: Despite being a completely faithful representation of the original game, with the same locations and descriptions, it left memory to spare.

[Mike] We targeted computers down to about 8K. I have a feeling we may have done a 4K version at some point. The text usually compressed to about 25% of its original size. [...] We realised that with the compression techniques we'd created, we could go way beyond [*Adventure*], and we created a whole endgame with about 70 additional rooms which weren't in the original. We tried to avoid "key and lock" puzzles where you need to get a key from one room and take it to another room. These get boring fast. We tried to be more inventive than this.

In the original *Adventure*, the game's finale sequence started when you picked up the final treasure and a "sepulchral voice

reverberating through the cave" announcing that it was closing soon. The player then went on a frantic dash to get out in time before the game ended. *Colossal Adventure* implemented a slight variance: Pete told Page 6 magazine in 1988, "The only thing that we actually moved is the food which we put in the forest. It was really because there was a lot of forest around, nothing actually to do with the game."

Whilst Level 9's first (and mythical) adventure game *Fantasy* sold a few hundred copies, *Colossal Adventure* sold a few thousand on the Nascom. A marked improvement, but it was not enough to sustain an income or be commercially viable. Because of the paltry returns, the Austin's felt more like hobbyists instead of participating in something that was going to earn a living.

[Mike] To give you a sense of scale, a brisk sales month was 11 games a week! The adverts were column line ads, four or five lines of text. They weren't terribly expensive. We made enough money from the sales to pay for the ads and a little bit more, so it was how it started to build. [...] I think the total sales on the Nascom was less than 100 [units] in total. £10 per product at the time was quite [expensive for consumers]. For a very small company with very small cash costs, it was nice. [It was] £1000 a year or something, [...] but sales only began to escalate when the mass market micros like the BBC and the ZX Spectrum came out.

The launch of the BBC Micro and accompanying ambitious Computer Literacy Project, along with Clive Sinclair's machines stoked the potential for Britain having a home computer mass market. More and more machines appeared, and the value of *A-Code*'s portability was clear – it would allow Level 9 to create games for every new computer, and being the first onto new machines brought the commercial advantages of being big fish in a small pond.

[Mike] The BBC [Micro] was a phenomenon, massively promoted and we were able to get games onto the [hardware] fairly early. Then when we began to get sales of 10 a week, so still not enormous, but you know, actually maybe there's something here we might be able to make a living out of.

Colossal Adventure retailed for 16K machines at £8 and £10 for 32K micros. Given that the average post-school teenager in 1981 or 1982 was earning £12 a week from a Youth Training Scheme, selling 10 computer games at £8 or £10 was a more agreeable way of making money.

[Mike] It helped that I was still at school at this point, so I had no real costs. My brothers, Nick, had just graduated from Imperial and started work, and Pete was working elsewhere. This was [still] very much a second job, a hobbyist king of thing this time.

Getting Acorn and Sinclair machines into schools and homes kickstarted a boom in software sales. Micro Adventurer magazine in 1985 speculated that *Colossal Adventure* eventually ended with healthy sales of 25000 copies. However, there were still other machines that were housed in schools around the country, including the RM 380Z that Level 9 experimented with.

[Mike] We did games on the 380Z because my school had one at the time and it was a simple port. It was a simple computer to program and generous on memory, so we took the Spectrum version and ported it across. It took us a couple of days, and I think we sold 2 copies, so wasn't the best return.

Incredibly, it's estimated that 20-30 micros were being launched, with 1983 regarded as the single busiest year for new systems. Andrew Morton in his "Amstrads and Ataris" book commented that Sinclair machines were "soon joined by a host of competition in the early years of the 1980s as machines were released from companies such as Acorn, Dragon, Tangerine and Commodore. They all wanted a share of a market that would soon be worth millions of pounds."

[Mike] It was a deliberate strategy to try and get to market first in some of these new computers. Having said that, some of them like the [Camputers] Lynx we were going to reject because at that point we were getting a little bigger and a little more choosy. They actually paid us some money to finance [ports of our games], I don't think it was huge, maybe £500 or something, but you know it was enough to say OK that will cover our direct costs so let's try it. Both the Lynx and the Oric were in that category.

The Austin's soon found that the BBC Micro was the standout, stable and capable computer to focus on as a development platform. Its forward-thinking and open-ended architecture allowed for upgrades, and hardware additions. Torch Computers (a Cambridge-based hardware company) released one of the first Z80 co-processors add-ons for the Beeb. It turned the machine into a workstation capable of running a version of the highly regarded *CP/M* disk operating system.

[Mike] We thought *CP/M* was a more stable, grown-up environment. They had floppy disks which was a big step forward for us at the time and they were relatively cheap. It's probably fair to say we spent most of our spare revenue in those first few years buying additional computers. There were so many computers coming out at that time, we needed the extra hardware to either develop on it or to be able to target it.

The flood of machines would prove to be the first real test for A-Code. The conceptual thinking behind the language wouldn't be fully realised until a machine with a completely different technical architecture appeared and was targeted. Would it work ported to another micro?

[Mike] We had to rewrite [*A-Code*] in 6502 assembler, because the Nascom was Z80, so that was a bit of a pain. But it didn't take that long because the *A-Code* interpreter was short, compact and the text compression [routines] were compact and so on. It was helpful to run the same Z80 code on the Torch [but] we had to rewrite the interpreter, the runtime and had to effectively write a driver library. The driver library was where most of the changes to the interface came in because on the Nascom it was a simplistic screen, whereas the BBC had more capabilities.

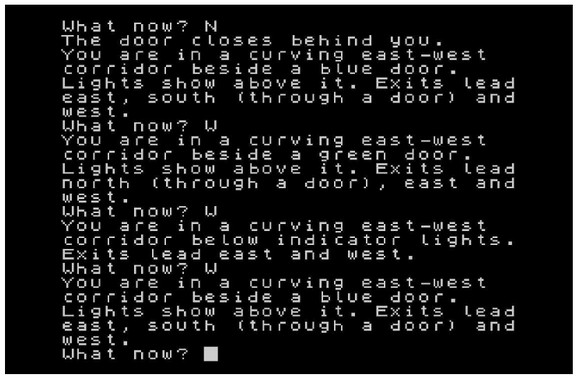
By December 1982 two sequels to *Colossal Adventure* arrived. *Adventure Quest* (a journey across Middle earth to find the Dark Tower and defeat the Demon Lord Agaliarept) and *Dungeon Adventure* (a perilous dungeon quest to confront evil forces) offered the brothers the first chance of writing a narrative beyond the *Adventure* mindset. Mike told Page 6, the Atari Users Magazine "*Adventure Quest* was the first game I designed from scratch and it's a very linear game. You go through about eight different zones, but you have to do them really in order [...] *Dungeon Quest* [...] is where you can wander around in a reasonable area and there are groups puzzles within that. If you get blocked on one group, you can go and try another area."

[Mike] Pete took on the design for the games. [...] I think he quite enjoyed the process, but I think he found it hard because his background was computing. [...] He dabbled with writing fiction at school and quite liked that. As a writer he developed and was good as time went on. [...] Even when the games [and] concepts were designed by others he always came along and tweaked and made the puzzles more fair. [That was] because quite often the game designers came up with something that was a nice idea, but nobody would actually understand.

It was Level 9 playing to each of the family's strengths and weaknesses. With Pete focusing on design, sister Margaret

[Colossal Adventure] Level 9's adaptation of Crowther and Woods' classic, and an incredible demonstration of programming genius. Their efficient A-Code adventure system squeezed the mainframe masterpiece into home micros with room to spare for an expanded game.

[The Archers] The collaboration with Mosaic Publishing led to several multi-choice parser titles for Vicky Carne's company. *The Archers* was by far one of their least successful games commercially, but is packed with witty text and clever plot lines.



[Snowball] The best of the *Silicon Dreams* trilogy, and perhaps the best 8-bit adventure from the Austins. Play heroine Kim Kimberly in this sharply crafted science fiction epic, boasting almost 7000 locations!

[Gnome Ranger] With it's razor-sharp humour, Ingrid Bottomlow's debut game underlined Level 9's reputation as one of the most versatile and original text adventure houses in the gnown world.

[The Price of Magik] Following *Red Moon*, *The Price of Magik* perfected the use of magic lore, and skillfully mastered the balance between Dungeons and Dragons, RPG and text adventuring elements.

[Scapeghost] Their last stylish adventure, showcasing some of the best 16-bit graphics of the time. *Scapeghost* is an intricate and atmospheric adventure with a haunting storyline of ghostly revenge.

joined her brothers as to take over the role of marketing and ramped up their distribution ties. Their father, John Austin, came out of retirement to become Managing Director and played a key role in steering the family's successes in those early periods. The two remaining brothers, Nick and Mike concentrated on games and adventure technologies.

[Mike] I'm not a writer in that sense. I can write functionally but I'm not a creative writer. I do understand technology and can build stuff quickly. I think that's right, you've got to understand different attributes and Nick was somebody who needed to have a clear target for what he was doing but then would go away and implement it very thoroughly. So, you know, we worked well from those points of view.

Marketed as three "Giant Adventures", *Colossal*, *Dungeon* and *Adventure Quest* soon had the *The Colossal Trilogy* nickname, and by 1984 they were being referenced as *The Middle Earth Trilogy* in advertising and by reviewers such as Peter Gerrard. It was a moniker that Pete Austin was in favour of as it "gave the players and buyers a sense of what to expect in the world". It wouldn't be until the Autumn of that year that Melbourne House made overtures for an official Lord of the Rings licence, so was there any thought at all of Level 9 doing it beforehand?

[Mike] We toyed with the idea but weren't big enough at the time and we didn't have the money really. We certainly were fans of Lord of the Rings, and we all liked the generic fantasy genre. *Colossal Adventure* hadn't sold huge numbers, but it was doing well and we enjoyed producing it and the feedback was great. So, we thought, yeah, let's do another one and then another one. I think it was more carrying on the momentum and then after three we thought, let's do something different.

So, the elves, orcs, dungeons, rings of power, sprawling deserts, wilder lands and mountains were soon left behind in the blackness of deep space. Their next game blasted into the science fiction genre with *Snowball*.

[Pete Austin] We wanted to do something different. There were a lot of sub-*Hobbit* adventures about, in which you went around slaying orcs. I was afraid the public would get fed up. If you keep doing games within a single genre, they compete with each other. Some people want to play every game with orcs and trolls or whatever, but most people play one and then move on for a while. So, it's good marketing to mix up your subjects. I read several thousand science fiction books when younger. My favourite authors included Bradbury, Asimov, Heinlein, Leiber, and Moorcock. More recently: Haldeman, Simmons, Gibson. My preference when writing is "hard" science fiction, that tries to be real and has a well-developed back story, as distinct from space opera or fantasy. If I had to pick one influence, then Heinlein.

Snowball, named after Snowball-9 the spaceship upon which the adventure is set, followed the journey of two million colonists in hibernation, headed toward their new home on a planet called Eden. During the transit, one of the occupants is unexpectedly awoken from stasis, and has to combat a member of the ship's crew that has turned rogue.

[Mike] Well, I rather like *Snowball* because it was different. I don't think there had been very many science fiction adventures by that point.

The woken colonist, an agent fortunately placed upon the mission to counter any threat to the ship, was perhaps one of the first female protagonists in an adventure game. She was called Kim Kimberly, named after a street close to where Pete was living in Bracknell. At the time, the game's press and documentation (in the most) used the

Kim Kimberley from *Snowball*. One of many Tim Nayce artworks that were included in early Level 9 packages.

gender-neutral Kim without reference to a pronoun.

[Mike] It was a deliberate choice, but we didn't want to create a character where the gender was used as a primary plot device.

[Pete] I thought it was fun to have keep the hero/heroine's gender unspecified until the end, because it didn't matter for any of the puzzles, and then reveal that she was female. One of the things that you can only do when writing for a text-only medium where the text is very short and the characterisation is largely in the player's mind.

The gender speculation laid to rest by the Austins was also clarified when the game's packaging referenced "Miss Kimberley's Wardrobe". There's no other reference within the game, and as Pete indicated it was therefore left to the player's own imaginations to determine. Regardless, the discussion of Kimberly is fascinating, especially when reading press interviews from the time. Like their vision for *A-Code*, the words and thoughts of the Austin's seemed ahead of their time in discussing the importance of genders, especially when it came to players of adventures in a perceived male dominated world of videogames in the 80s.

Crash Magazine speculated that a third of Level 9's sales were to female players, and Pete emphasised the importance of that demographic by commenting that the "main way to make women more interested in computers is to change the advertising. [...] It [is] belter to point out that computer programs can be interesting, informative and broaden the mind."

In an ever-changing world there was the appreciation that traditional narratives could also take them into areas of controversy.

[Pete] This was the problem. Lots of people sent us a one-page idea for a game, but when we said, "That's really great, but we need it 10x bigger, and then we'll pay for the design", almost everyone dropped out. Very disappointing because it should only have taken a few weeks.

[Mike] Most of them were just ideas with no real substance. Creating a full game design is a lot of work – comparable to writing a novel.

One submission that did catch their attention was written by a Level 9 adventure super-fan, Sue Gazzard. She had purchased a BBC Micro and played adventures solidly for three months. "I've been known to stay up till 2am. I found adventures fired my imagination," she confessed to Micro Adventurer. Sue's theme pitted the player against a group of evil timelords who were attempting to rewrite history. Guided by Father Time, the player had to travel through nine different eras—from the Ice Age to the far future—collecting nine symbolic objects to restore the timeline to its correct path. It all seemed very similar to a very well-known Science Fiction television franchise that potentially could be licenced.

[Mike] We were working with Vicky Carne [of Mosaic Publishing – see Issue 03] who was very much from the traditionalist publishing arena and was exploring lots of licences. I think she came up with that as a possible licence along with a number of others but it didn't end up happening.

To give it a breath of originality, Mike and Nick took Sue's ideas and handed it Pete to tinker with. After the addition of several new puzzles they were delighted with the result, releasing the game as the brilliant *Lords of Time. Lords* was revered by many journalists,

"We had to add graphics because competitors were starting to do so. We were trying to create a picture in less storage than is used today in the header of an email"

An interview in Micro Adventurer magazine revealed that Level 9 struggled with racism and sexism in traditional stories and fairy tales, dropping a plan for a game called *Island* in the process. It was quite radical thinking for the mid-80s.

[Pete] I wanted to get close to 50% male/female lead characters, because that seemed fair and more fun. This got a bit exaggerated because other authors were sticking with the male-hero archetype and so journalists tended to include it.

Snowball was also a game that courted controversy because of its bold claim of containing 7000 (yes, 7000) locations. There was a general industry preoccupation with statistics across all genres. It always looked good on boxes and adverts, but it wasn't necessarily an indication of quality - as John Wilson and Zenobi proved with his single-location *Behind Closed Doors* series. We were in the infancy of the industry.

[Mike] We wanted to give value to buyers, and this was a good rule of thumb. There was a bit of a war between adventure game creators at the time about the number of rooms, and we wanted to set a number that would be very hard to beat! They were genuinely 7000 locations, although the descriptions were – to say the least – rather repetitive. [...] [It] was a bit of a gimmick, and as long as you didn't get into an endless loop of exploring those it didn't get in the way too much for me. It did help with publicity.

During the development of *Snowball's* sequel, many unsolicited game designs began to land on the doorstep of Level 9, including one set in the Bermuda Triangle that the brothers liked, but ran out of steam attempting to progress the design.

including The Micro User's Mad Hatter [Bob Redrup's pseudonym] citing the game as one of his favourite titles.

[Pete] Sue produced a nice game design, which was different from my ideas. We were soliciting ideas at the time. She sent a letter with the design on paper – printed I think. I got her to extend it, then added several more puzzles myself.

[Mike] We spent quite a long time working with Sue and going through iterations. Then we did more work on the puzzles. It was a rare occurrence to receive a submission worthy of converting into a fully blown commercial release.

Working through the various design stages was a laborious and drawn-out process. An eon before the invention of the internet and email, working with freelancers and external parties provided time consuming and logistically difficult.

[Mike] It was snail mail, I mean, everything was. We worked with quite a few freelancers, both developers over the years and artists and some game designers like Sue. We'd post things, and she'd post back a version of the design. We'd go back with amendments and she'd come back with a revised one. It was actually more challenging with some of the graphic artists where we'd have to talk to them on the phone to say what we wanted or we'd post them design briefs.

Sue had originally planned to write three games for Level 9. After *Lords*, her second design, called *Neptune* was deemed too one-paced. Despite Mike and Pete attempting to steer the narrative into a more rounded game it never reached a suitable point where it could be released. After the disappointment, Your Computer magazine's "Beating The Yanks At Their Own Game" feature in July 1985

LEVEL 9 IS THE MAGIC NUMBER

The Austins wove personal and political references into many of their games, including the addition of the number 9 - for obvious reasons. *Lords of Time* is one example that has 9 levels.

reported that Sue moved on to work in conjunction with Roy Carnell [working as an independent after the collapse of his own Carnell Software label]. Together, their new adventure "epic" was to be marketed by Mastertronic for their new premium Mastervision brand, but it never materialised and disappointingly Gazzard didn't appear on any subsequent game credits.

Return of Eden was the long-awaited continuation of *Snowball*. Through exploring this source material Level 9 showed their versatility and began to establish themselves as one of the great adventure writers of the 8-bit era. Thomas A Christie in The Spectrum of Adventure remarked "*Snowball* and its sequel[s] would be the closest that [Level 9] would come to replicating the entertaining the complex sci-fi environment of Steve Meretzky's legendary Infocom duology, *Planetfall* and *Stationfall*." It was worthy praise indeed.

Return to Eden [featuring Mike's favourite artwork], found hero Kim marooned on the deadly paradise of Eden after a dramatic turn of events since the end of the first game. It seemed that she'd been blamed for the damage caused to the spaceship and was exiled to the planet's surface awaiting execution. The game saw the player attempting to help Kim escape from the dangerous faun and fauna of the planet, as well as its deadly robotic population.

To complete what would become *The Silicon Dreams Trilogy*, Level 9 released *The Worm in Paradise* later in the same year – albeit a century later in the timeline from the preceding two adventures, and thus without Kim Kimberly.

Snowball was a text-only adventure, but with *Return to Eden* and *Worm in Paradise*, Level 9 tweaked *A-Code* to allow the inclusion of location graphics for the first time. As with the location count, the inclusion of graphics was due to the team bowing to commercial pressure. For most adventure writers it was an unwelcome one.

The balance between the quality of graphics and depth of text was difficult to achieve. Many authors either sacrificed location text or created games with multiple parts to fit images alongside a decent sized adventure. For cassette owners it was tiresome and laborious having to load each section. The Austin's took a different tack, aiming to keep games to a single load. They continued to adapt *A-Code*, with each game release featuring an incremental, but largely-backwards compatible engine, and they managed superhuman compression rates of text. Later versions of the compression system got this down to a miniscule 15% of the original size.

[Mike] *[A-Code]* was consistent from start to finish. The thing that changed a lot was the text compression. The big change around this time was the change from compressed to squashed, which was also when we moved to this idea of having the parser words included in the main dictionary. I think that happened in terms of changing the development environment.

The graphics quality of the plot, line and fill images produced were suspect and not up to the visuals seen in competitor multi-load games, or in adventures such as *Heroes of Karn* from Interceptor Micros [see **Terry Greer - Issue 04**] who sacrificed the number of graphics for a higher quality bitmap-style image.

[Mike] We had to add graphics because competitors were starting to do so. We were trying to create a picture in less storage than is used today in the header of an email, so we created the minimum graphics that would create the impression we were looking for.

[Pete] We had a portable programming system, *A-Code*, but it turned out to be very difficult to produce similarly-portable graphics in the space available. Digitising artwork didn't work well enough because these home computers had too few colours and fully digital art based on polygons was very labour-intensive.

[Mike] The reason we went for vectors was that they were incredibly tiny in terms of storage. Each of the graphics was maybe 100 bytes, if that. [...] On a 32K machine, I can't remember the exact breakdown, but something like 4 or 5K of code, 8 or 9K of text and a few other bits and pieces, parsing directories [and so on]. The budget for the graphics was maybe a K for the entire game, just tiny.

The period between 1983 and 1985 was perhaps Level 9's most productive era. They released 12 games and pushed the boundaries of what *A-Code* could achieve. The structure of the company evolved to meet the growing professionalism and demands of the market, including the presentation of their product. From the start they'd developed a trademark brand of packaging, at first with Ziploc, homegrown packages, but then changed to produce large oversized cardboard boxes with line artwork.

[Mike] We used Ziploc bags because the volumes were so small we couldn't afford to do anything else. [...] We went onto the black and white cardboard because we got an order in [September 1983] from a company called Microdealer. [...] They said we really like the games, and we'd like to order 5000 please but for delivery by the middle of October – certainly to get them to shops before Christmas. So, we took a big gulp and figured out how we could do that.

Delivering 5000 units proved to be a monumental task. The British computer game ecosystem was just evolving from bedroom programmers selling directly through mail-order ads in magazines, to high street stores putting stock onto shelves. Packaging and cassette production had existed for years for music, but for videogames, and the nuances that computer audio required, it was still in its infancy.

[Mike] We got a local cardboard company lined up, and Pete designed it in conjunction with their cardboard folding person, and we got one of our artists to do the design. It was a very geometric design [since] it had to be done extremely quickly. We wanted something to be high impact and would stand out but could be built really quickly. [...] There were some early tape duplication companies, but they were mostly rubbish, so we bought 10 high quality tape decks and lots and lots of TDK cassettes because they were by far the best quality at the time. We could duplicate one master onto 10 tapes at a time, and it was workable to produce that kind of volume if we ran continuously for a month. I started ferrying the boxes out to my school friends, and they'd fold them. I'd go back a few days later and pick up a car full of boxes ready to have games put in.

Later releases featured superb cover and illustration artwork by Godfrey Dowson and included many additional items offered to give customers a better deal – such as posters and hint sheets. Their cottage industry duties of creating everything to do with packaging and fulfilling cassette duplication was behind them. The cardboard boxes changed to oversized vinyl, and the disappointingly, into the industry standard "clam case".

[Mike] It was as retail matured, so shops started to push back on the big format. We went briefly to the plastic sleeve ones [...] and then the retailer said we don't have the shelf space if you send it in that format. So, we had to move down to the smaller format. It was a shame, but it was the commercial reality – we wouldn't have got stock into retailers if we hadn't done that.

The high sale demands of 1985 meant Level 9 took on extra staff and

[Right] Artist Oliver Frey captures the fearsome life on a distant planet in this *Red Moon* inspired cover for Crash Magazine.

contracted external writers to maintain their productivity. *Emerald Isle* was the first game to take advantage of a dedicated external freelancer, Shaun Abbott, leaving Pete with time to work on another seminal title with another new member of the team, David Williamson.

Pete and David's new game, *Red Moon* was pitched as a hybrid adventure. It had strong elements of their Dungeons and Dragons heritage, with combat and role playing (an introduction of a health system or "hit" points), combined with a highly developed system of spell-casting (the CAST ZAP command for example). It was another standout title, winning best adventure of the year in a slew of magazines including ZZap!64, Amtix, Crash and C&VG.

With their reputation at its zenith, it was unsurprising that other publishers knocked on the door wanting to make use of their technology and creativity. Virgin Games and new start up publisher Mosaic, headed by Vicky Carne, contracted Level 9 to work on several licenced titles.

[Mike] Commodore were doing a big push on the 64 and they wanted a to do a big bundle deal with lots of games included in the pack.

Sue Townsend's hugely popular books catapulted teenager Adrian Mole and his secret diary into the imagination of children and young adults up and down the country. It was perfect fodder for a Level 9 text-based game, and so *The Secret Diary of Adrian Mole Aged 13 1/4* was released by Mosaic Publishing in 1985 and *The Growing Pains of Adrian Mole* by Virgin Games in 1987. Both games dispensed with the standard parser-based verb/noun experience and opted for an input mechanism that offered multiple choice pathways instead of typed commands.

[Mike] The vision was to have interactive novels, and to aim to appeal to a wider market than the puzzle-oriented audience for adventure games. [...] It was simplified in terms of gameplay because Commodore was [selling] to a mass-market, so they wanted something that was very accessible for people. We had some discussions about it being an adventure with puzzles, but it would have been difficult to get the same flavour [as the books].

With a streamlined storyline, the team could concentrate on the design and writing rather than producing code. Their *A-Code* language proved a flexible and adaptable platform, and they had little problem making minor enhancements to the functionality to enable pre-determined pathways.

Mole was a return to taking someone else's work, as with *Adventure*, and adapting the content for a brand-new audience. Pete told Popular Computing Weekly that he'd "done an awful lot of research on Sue Townsend. I had to find everything I could about her because I had to copy her writing style."

It's ironic, and an indication of the state of the games industry that their least creative project in terms of intellectual property and story writing became their most successful with a rumoured 165,000 sales.

[Mike] We worked very closely with Sue Townsend and I understand she was very pleased with the result. [...] I think *Mole* actually appealed mainly to older people than our traditional audience. Mole was so hugely successful – by far and away our best-ever selling game, [but] a win is a win, it would have been nice if some of the other games had sold that much.

[Pete] My main memory of *Mole* was working 20-hour days, because the release was brought forward by several weeks to tie in with [the Commodore] joint-marketing deal.

[Mike] As a writer [Pete] developed as time went on, and with *Adrian Mole* we actually wrote all of the text. Sue actually said that she thought some of Pete's text was better than hers, which is a really big, big compliment.

That "huge pressure" and crunch that Pete experienced had an impact on the quality of the game and left some areas that could have been made better. Perhaps it was an indication that this type of developer/publisher relationship could affect their creativity and cloud their previous non-negotiable high standards of quality. One positive outcome from *Mole* was a piece of new *A-Code* system technology that enabled computers to display text and graphics to the screen at the same time.

Mike told Popular Computing Weekly that "in *Mole*, we have two tasks running at the same time; one is playing the game, the other is controlling the screen. There's a third task on the Spectrum. Amstrad and MSX, controlling a 100-character buffer. This means you can play the game in picture mode without taking hours."

Until now, adventure games generally displayed a location graphic before a pause, a wipe of the screen and then the description, but because of the text-centric nature of *Mole*, the buffer developed by Mike streamlined the experience for players.

[Mike] It went in as a standard feature on all the platforms. We had to implement our own multi-tasking system since there wasn't really an operating system on most of these home computers. [...] We had to write our own BIOS, so we implemented a timeshare system and one of the main applications was dividing out the graphics from the text in passing. It added a lot of complexity to the platform part of the adventure system.

Radio 4's long running radio serial *The Archers* (another multiple-choice game for Mosaic co-written by the BBC's scriptwriting team) was next, and *The Saga of Erik The Viking* based upon ex-Monty Python Terry Jones' book completed the firm's work-for-hire licenced line-up.

[Pete] [*Erik The Viking*] was great and we based other puzzles on Viking myths. We met Terry in London and went to the Yorvik Viking centre for research. The pictures turned out great in my opinion. Very happy with that game.

The last game in this golden era for the company was *The Price of Magik*, written by Pete and David Williamson. They returned to the fantasy worlds of the original trilogies but dispensed with some of the traditional rules constraining adventures. Set in a magical equivalent of earth, *Magik* expanded on the magic systems of *Red Moon*, and saw the player as a novice spellcaster attempting to become as powerful as crazed sorcerer Myglar, in order to defeat him and recover a powerful crystal.

New ideas breathed life into *Magik*, and the need to solve all the puzzles before completing the quest was removed. It was just one example of Level 9 trying to disrupt the genre.

The late 80s gaming landscape was continuing to evolve at such a rapid rate, and the 8-bit market, especially cassettes, was in perpetual decline. A new direction for the company, and *A-Code* was needed. Richard Hewison's history of Level 9 observed that the company "employed the services of John Jones-Steele to write for them a new adventure writing system for their next generation of adventures." Steele, a veteran of adventure writing was engaged to "devise a system that would allow for 'real' characters to inhabit the game worlds they created and allow people to recruit them and have many characters performing the same task at once to solve certain puzzles."

[Mike] It wasn't really a new version of *A-Code* as such – just new capabilities for the core language and system. The non-player character handling was rolled into all the games.

[Pete] We did [implement] a complicated NPC system based on what we called *Racetracks* so NPCs kept doing stuff even while you weren't there and a system for giving commands to NPCs so they could go off and do stuff.

Non-Player, or computer-controlled characters up until that time had followed very limited scripts or rule-based behaviours. It was an early implementation of Artificial Intelligence for text adventures that attempted to make the worlds feel inhabited and interactive for players.

[Mike] *Racetracks* was about the particular logic driving NPCs in parallel [with the player's actions]. We made it so NPCs could do everything the human player could do, so they effectively were executing the same set of commands as humans would.

The concept had appeared in a few British games before. Legend's *Valhalla* and Trevor Hall's *Twin Kingdom Valley* both implemented non-playable characters with interesting, but restricted capabilities. Even Level 9's own *Fantasy* had computer-controlled entities with some semblance of artificial intelligence. Pete recalled to Sinclair User, "There were a lot of characters wandering around who changed according to your actions." However, it was *The Hobbit* (and subsequently *Sherlock*) by Melbourne House that transcended all before, making characters appear to have a true mind of their own, and in frequent cases exhibit emergent behaviours.

[Mike] [Our NPCs] had some heuristics about what they were going to do, and those were some specific kind of sequences they did at particular times. If an event happened it would trigger an NPC to start, what I'd guess we'd call a quest – where we could go to that place and pick up the key and go back to the safe and unlock it. That was also available to the user in the parser as a compound command, so it was working internally and externally.

Implementing NPCs didn't come without challenges. Programmers had to devise clever ways to allow computer-controlled characters to replicate human decisions and choices. Within *A-Code* one of the specific barriers to overcome was route finding. The simple operation of allowing a character to navigate their way around the game's map.

[Mike] Nowadays you'd use one of the route-finding algorithms which essentially build a tree of possible routes, you know, just like a GPS. On the systems at that time it would have been just about possible but would have been slow in execution. So, the *Racetracks* were a way of NPCs knowing a particular route that would get them from various A's to B's.

Abandoning 8-bits and concentrating on development for the new 16-bit computers and consoles was going to take financial clout, both in terms of technology and the rise in marketing spend. The Austins needed an injection of capital and support to bring bigger games to market. Telecomsoft, the software arm of British Telecom had created an offshoot label called Rainbird headed by Tony Rainbird. Tony had the ambition of expanding into US markets with the development of high-quality 16-bit titles. He saw Level 9 as the best adventure writers in the country and was keen to bring them on board to provide new products.

[Mike] Tony got in touch and met us and waved around a big number to what we'd make for the older games and new series. It was very appealing. We saw that we could potentially get mass distribution and those games [into the] mainstream [since] BT were putting a large amount of money into the promotion and marketing of [Rainbird]. [...] They were able to pay us a large advance – much more than we were able to get from selling games directly.

The agreement with Rainbird was announced at the 1986 Spring Consumer Electronics Show at Olympia in London. Tony Rainbird and the rest of the Austin's were in attendance for the high-profile launch, adding colour to what otherwise would have been a quite drab event. Pete told ZX Computing Monthly that "[our] agreement marks a step forward in adventure games. We expect a substantial Increase in adventure playing and Rainbird are prepared to support our products in a unique way."

Level 9 were to produce four titles spread across 8-bit and 16-bit formats, including updated versions of three existing Level 9 trilogies. *The Middle Earth Trilogy* was re-released as *The Jewels of Darkness*, and *Snowball*, *Return to Eden* and *The Worm in Paradise* were bundled together as *Silicon Dreams*. Both were given the full Rainbird treatment, with lavish oversized packaging, and the inclusion of a novella written by Peter McBride.

[Pete] Peter McBride is a good author. The games were very limited in size, so the obvious way to make them more immersive was to include a novella about the back story, which he wrote and I edited. Some other publishers were doing the same.

As well as the inclusion of a novella the games themselves were all revisited with several plot tweaks, more atmospheric text and a host of features including enhanced parsers and graphics on the new 16-bit versions.

[Mike] [The new] 16-bit [versions] didn't really make much difference to us; it was the same *A-Code* that ported to both. [...] The text compression we were using at this point was much, much slower than the first version. In terms of runtime it was comparable, but in terms of the compression phase it was multi-phase and doing a lot more analysis in order to get better efficiency.

The Austin's realised that to evolve for the 16-bit market they had to reconsider the direction of their adventures and their underlying code structure. Whereas *A-Code* was built to overcome the limitations of British hardware, STs and Amigas shipped with a minimum of 512MB of RAM and the ability to use disk storage meant that their early ideas for larger and more expansive games could be explored. There was also the emergence of a new adventure house making huge waves in British waters - Magnetic Scrolls.

The potential of the new machines, and the prospect of a more competitive market led to the start of a new direction for *A-Code*. It was heavily refined to create their next game, *Knight Orc* - a multi-character, single-player game aiming to turn the tables on Orc-bashing in traditional fantasy novels and games.

Unfortunately, the literal acronym for *Knight Orc Adventure System (KOAS)* didn't match the cooler sounding and more legible *KAOS* acronym which contained the same letters. Mike credits that to the smart marketing department at Rainbird. *KAOS* was, it seems just a further iteration of *A-Code* not a brand-spanking new system. It was modified to include digitized Godfrey Dowson graphics and emulated the experience of multi-user text games – something that they'd thought about with an earlier idea for a radical networked game called *Avalon*.

[Mike] *Avalon* [was] probably something too ambitious. We looked at a network of [Amiga] computers with modems, and we looked at a telephone-based game using premium numbers to phone up and play. We played around with both of those ideas. [...] We did a design brief, and some sort of technical feasibility studies but I don't think it got much beyond that because we worked out that the amount of capital involved to get us off the ground would be too high and too risky. It was also around the time when there was a lot of bad publicity around premium rate phone lines, and I don't think there was an alternative way of doing [it]. We couldn't see how to make it work, so we dropped it.

Avalon set out to change the way that MUDs were perceived, with a rich backstory and a reported 10,000 locations and 1,000 computer-controlled characters. With *Knight Orc* they set out to change the way text adventures were perceived. *Snowball's* Kim Kimberly introduced the protagonist as a female. In *Knight Orc* they set the protagonist as one of the genre's traditional adversaries, a rather unpleasant Orc. Set across three parts of varying difficulty, *Knight Orc* was published on the ST and Amiga but ported back to the 8-bit market soon afterwards.

Jimmy Maher in his *KAOS* blog gushed praise: "Boy, was it original. While Magnetic Scrolls was polishing up a more perfect *Zork* in the form of *Guild of Thieves*, Level 9 was seemingly trying to blow up just about every assumption ever held about the genre with *Knight Orc*. It all added up to the most radical single reimagining of the text adventure of the genre's commercial era. Infocom had played with more dynamic, responsive story worlds of their own, particularly in their first trilogy of mystery games, but never on a scale like this."

Given more time, and with the right backing from a patient publisher *Knight Orc* promised much, especially with the potential evolving capabilities of the *KAOS* language. Through Rainbird, that potential would never be realised - the relationship broke down when the man behind the label, and the driving force behind most of the products left the company.

[Mike] Everything changed when Tony Rainbird left. He was the man, as we saw it, who could get things done.

With Tony gone, and the Rainbird agreement in tatters, the planned final compilation of adventures – *Time & Magik* (*Lords of Time*, *Red Moon* and *The Price of Magik* complete with upgraded parsers and improved/added graphics) was dropped and Level 9's contract was terminated by mutual agreement. Richard Hewison [**see 2025 Special Edition Issue**] speculated that the package was in the final throws of being released.

[Mike] The separation was pretty sudden as I remember. Essentially British Telecom lost patience with the idea and decided that software wasn't the way of the future. They progressively shutdown Rainbird so new titles were not going to happen. I don't know the whole story, but that's certainly what we were told. [...] I think it got to the point of pre-promotion, but I don't think it was ever actually launched or manufactured.

Without Tony, the Austin's felt that they had lost a valuable ally within the company. After Rainbird, Tony setup Special Reserve, a discount mail-order company that attempted to invigorate the text adventure community with a dedicated magazine called Official Secrets. Though Magnetic Scrolls produced a specifically commissioned game (*Myth*) for the project, contrary to speculation, Mike doesn't recall Level 9 being involved.

So, what really happened at Telecomsoft? On one side, corporate management claimed unacceptable delays in development. On the other, rumours swirled in the gaming press that BT had upset the Austin's by spending more time and effort marketing and promoting Anita Sinclair, and the adventures of her other Rainbird text

[Left] Ingrid Bottomlow, the intellectual-but-clumsy gnome from *Gnome Ranger* and *Ingrid's Back!*

A-CODE ARCHIVE

In 2022, text adventure author and videogame preservationist Stefan Vogt approached Mike Austin to release *A-Code* and the Level 9 archive to the community. After a long process of digitising his content of around 1000 disks and a myriad of ephemera, Mike announced in 2025 the availability of a public archive.

The repository is already crammed with source code, games, documentation and the promise of more material to come. You can access the Level 9 archive here:

github.com/MikeTheTechie/Level9-Public

[Above] Police officer Alan Chance attends his own funeral in the deliciously named *Scapeghost*.

adventure stablemate – Magnetic Scrolls.

[Mike] No, no. I'm not sure we knew the name Magnetic Scrolls, but we knew there was going to be another adventure company [alongside us]. Scrolls only appeared with *The Pawn*, [and] we certainly weren't aware of them until that.

It was difficult to come to a conclusion to why the relationship with Rainbird wasn't the great success it should have been. The egos of the companies involved played their part, but it was also a mistake for Level 9 to spend so much time re-engineering their existing 8-bit games for 16-bit computers instead of progressing thier adventure technology. The majority of Amiga and Atari ST owners had upgraded to the new machines from Spectrums and Commodores, so they had already played and owned Level 9 games before. There was withering criticism over their decision not to revamp the graphics routines for the new micros. Chris Jenkins in PCW magazine was scathing in his Rainbird supplement, saying that the "graphics [...] would look fine on a *Quilled* adventure on the Spectrum or CBM64, but are a positive insult on the ST."

[Mike] I don't know, we were able to do more with them. It's a common pattern with consoles that games companies go back and re-implement some of their earlier games using the newest technology and that's what we did. It's fair to say the update of the *Jewels of Darkness Trilogy* took massively longer than we expected. That was a surprise and it was a very cumbersome process.

Delays were nothing new. There had been previous long silences between releases. A year had passed between *Silicon Dreams* and the next release which prompted Pete to use his column in Adventurer's Club magazine to respond to the conjecture. The way that Level 9 wanted to work irked with the Rainbird suits. "Unavoidable delays" were cited in the development of *Time & Magik*, and Richard Hewison in his Retro Gamer article recalled "grumbles from the publisher's side on the number of bugs being found during the development of *Knight Orc*."

[Mike] [With Rainbird there] was more focus on a preconceived idea rather than coming up with the right answer for the game, if that makes sense. Working with a big corporate like BT, they had a very rigid structure for how everything was going to work. You had to come up with a design, and an implementation and then you had to test the implementation to match the design.

The straitjacket of formal development processes didn't suit Level 9's inherent way of creating games. They were pragmatic as an independent developer, and they'd evolved *A-Code* to a point where they knew it worked, so didn't have to go through the box ticking exercises of testing each and every version on different platforms.

[Mike] Testing was fine and fair enough, but some of the requirement to get to a particular design point [was frustrating]. We'd develop in a more agile way, iterating rapidly and you develop to the right solution rather than to some preconceived idea. Telecom came from a background where they worked very much in waterfall design, so it had to meet the spec, otherwise it was wrong. Even if it was better, it didn't matter, it was still wrong. So, you'd have to go back and do it again.

Now in limbo, their next game in development, *Gnome Ranger* was marketed as an independent Level 9 release in 1987. Subtitled *The Journal of Ingrid Bottomlow* it was another departure for the company, this time embellishing the small amounts of humour found in their other titles, to a game that was a fully blown comedy from the start.

[Pete] [I was] just mixing it up. I like silly stories too. This started as a new take on "Cold Comfort Farm" [a comic novel by English author Stella Gibbons].

It was a return to the classic packaging of the early days, with a big, bold sized box, emblazed with the Level 9 logo and filled with posters. Like *Knight Orc*, it was ported back to a selection of 8-bit computers (albeit without graphics on non-disk drive systems), including a conversion to the commercially obsolete BBC Micro. Perhaps this was the ultimate gesture by the Austin's to provide just how adaptable and portable *A-Code* was?

Gnome Ranger split the opinions of the press: Some loved the humour, while others were not able to grasp the surrealism and the addition of other jokey elements – such as prefixing every word beginning "N" with "G", such as GNORTH. The Games Machine said "perhaps it's the plot, it is so far out of context with reality that it

cannot possibly be believed", whereas ACE magazine wanted more of the same saying "*Gnome Ranger* has improved on *Knight Orc* in the puzzle stakes, atmosphere, plot, and the characters are more interesting."

Having severed their ties with Rainbird, Level 9 secured an exclusive distribution deal with a new label called Mandarin in May of 1988. It was the brainchild of former Micro Power and Superior Software marketing guru Christopher Payne bringing together Red Rat, The Powerhouse, Bubble Bus and French outfit Jawx International under one brand. Mandarin picked up where Rainbird had left off and published *Time and Magik* – the trilogy endlessly delayed under the London outfit.

The original *Lords of Time* adventure had been advertised as being the "first in the Lords of Time saga". Further original games in the series never materialised, so it was *Red Moon* and *Price of Magik* that were cleverly woven together with *Lords* to form the new compilation trilogy. Unlike the *Silicon Dreams* trio, *Time and Magik* was more a continuation of a theme than continuation of a storyline.

Mandarin was offered *Lancelot*, a game based upon the legend of King Arthur and another nod to the interest in the abandoned *Avalon* project

[Mike] To some degree. It was a couple of years later, but some of the research went into it. I know at the time we ended up with a large number of books on Lancelot and Knights of the Round Table.

Lancelot contained a series of puzzles that linked to a real-life prize. A replica of the Holy Grail was produced, made of solid silver, semi-precious stones and plated with gold. Chris Payne was keen to boast about the value. "I think we paid about £2,500 for it to be made, so with a shop mark-up of 100% we felt that we could promote it as being worth £5,000." he commented on his website.

Around seven months after the game's release, in June 1989, a man called John Sweeney [**see 2025 Special Edition**], a Systems Analyst with IBM and a lifelong Arthurian buff solved the puzzle and collected the prize. In a magazine article he explained how he beat the game, "I have always been interested in legend and myth, and [...] this competition gave me the chance to combine the two, but towards the end the clues were very hard and at times obscure."

In the end, it transpired that Mandarin was no place for the Gnome, and Bottomlow's second appearance *Ingrid's Back!* previewed at the 1988 PCW Show and was back as a self-published title. *Ingrid* demonstrated that Level 9 were beginning to get into their stride with the *KAOS* evolution of *A-Code*, ironing out a lot of the bugs that had plagued *Lancelot*. They mastered the balance between humour and twee-ness of the narrative in *Ingrid*, and the NPCs, often confusing and shallow in *Knight Orc* and *Gnome Ranger* now came to life bursting with character and believable behaviour.

Ingrid was certainly without doubt the most enjoyable and accomplished adventure since the *Silicon Dreams Trilogy* and took the runner-up prize at the 1989 Golden Joystick Awards. Despite the plaudits, sales were very disappointing. Reportedly, *Gnome Ranger: Gnome Free* - a third instalment was designed and written, but coding was never initiated on any platform.

[Mike] I do remember there was a design, but in terms of implementation I don't think we got terribly far. But I may be wrong. At the moment I haven't found any disks with it on.

In late 1989 at the PC Show, Level 9 announced to a stunned press and gaming public that they were exiting and leaving the traditional text adventure market behind. The wonderfully titled *Scapeghost* would be their last adventure. Pete told ACE magazine "We shall not be releasing any more adventures unless Scapeghost sells much better than expected. [...] Nowadays everyone wants animated graphics and arcade action."

[Opposite] Godfrey Dowson's evocative *Scapeghost* artwork that was also included as a poster in the game packaging.

In development as *Things That Go Bump In The Night* and then *Spook*, *Scapeghost* was an original game based upon a concept by veteran adventurer Peter Gerrard with assistance from Sandra Sharkey. The narrative was first pitched to Level 9 in August of 1988 and the plot followed an undercover police officer Alan Chance, infiltrating a drugs gang. The gang discovered Alan's identify due to a mistake wrongly credited to the officer, and killed him, taking a colleague prisoner. Akin to a famous Hollywood movie storyline that would follow a year afterwards, Alan is given a chance of redemption and is returned to the world as a ghost, given three nights to clear his name and gain revenge.

In terms of pure craft, *Scapeghost* didn't reach the heights of *Ingrid* and showed an underlying disenfranchise with the current adventure landscape from the company. But it was of their best games, with a competent mixture of witty text from the writers, and a title that showcased the pinnacle of *A-Code* technology, especially on 16-bit machines where its complimentary graphics were of such a high quality that they finally were comparable with Magnetic Scrolls' output. In the end, money talked, and sales continued to decline despite the rising quality of games. *Scapeghost* reported to have sold a disappointing 15,000 copies.

[Mike] I think it was actually half of that, about 7000 [units]. [...] We had a discount of 50% [for retailers] so it was not really viable by that point, we just weren't selling enough. It was a great pity, because I thought it was probably our best game. I like the plot and storyline, and I think it was a very sophisticated implementation.

The final iteration of *A-Code*, moved from *KAOS* through a monumental evolution into a graphics-oriented RPG *environment* firstly named *HERO*, then christened *HUGE* - the *wHolly Universal Games Engine*. Level 9 had travelled full circle to the *Q1* language originally devised in 1979 to make portable arcade-action games with a D&D element. After initially dismissing the idea of icon-driven games (Pete once commented that "they don't give enough flexibility, and they get rid of too many of the puzzles"), they soon embraced the new graphical interface with Mike explaining to Retro Gamer magazine that "the way we were heading was very much like the way things were being taken forward by Sierra On-Line in America."

Sierra On-Line, who Mike referred to were producing atmospheric point-and-click graphical adventures. They were joined by fellow US developer Cinemaware who created games that placed the emphasise on action, graphics and ease of play. Along with LucasArts they were sharking up the traditional adventure market.

Nine Lives and Nine Green Murders was produced in-house as an ambitious animated detective adventure, put together to demonstrate to publishers the capabilities of the *HUGE* engine and to counter the American threat. The design was inspired by contemporaries (like Sierra's *Police Quest*) but expanded with large animated sprites, 3D-style environments, background music, digitised sound effects and a menu-driven interface.

Former Mirrorsoft staffer John Cook had just started working as an agent for Level 9 and had ties to Cinemaware given that his previous employer was now acting as European distributor for the US outfit. He was able to setup a meeting to discuss a potential deal for *Nine Lives*, but Cinemaware were more interested in finding someone to produce PC and Atari ST ports of its recent Amiga hit *It Came From The Desert*. Level 9 won the conversion work, and were also able to come to an agreement to publish *Nine Lives* which, after a year of further development was renamed *Grange Murders*.

Cinemaware soon ran into financial difficulties. Piracy was taking a toll on sales, and they had been late to the PC market, investing millions in CD-ROM and FMV technologies that they were unable to claw back. The collapse of the US publisher meant the end of any further development on *Grange Murders* and Level 9's final release was *Champion of the RAJ*, a half-hearted *Defender of the Crown*-esque RPG commissioned by PSS/Mirrorsoft. Magazine reviews were uncomplimentary. Their inexperience in the genre, and the infancy of their technology was telling.

THE LOST LEVELS

Along with *Gnome Ranger 3*, several other Level 9 games remain in the wilderness. After *The Archers*, the Yes Minister TV tie-in with Mosaic was passed upon and Oxford Digital Enterprises took over development.

The rumoured Star Wars parody, *Bizarre Wars* or *Gross Encounters of the Worst Kind* written in conjunction with Fergus McNeill and Delta 4 failed to make it further than an initial design. On the prospect of working with Delta 4, Mike said "[It would have been] a lot of fun! Fergus was so inventive and rather anarchic in his writing, a good balance for our tech."

One licence which eluded the brothers was Discworld. "Yes, we got some way towards doing a Terry Pratchett licence" muses Pete, "It didn't come off, and that's one of my big regrets. I think it would have been great!"

Another *HUGE* game, *Billy the Kid* for Ocean Software was reportedly abandoned. Amiga Power magazine in their October 1992 issue featured *Billy* in a special about unreleased games. It contained a quote by Ocean's Software Director, Gary Bracey, regarding the game's status: "It was not completed to our satisfaction, and it wasn't fully debugged. It reached a point where we decided it wouldn't be released." Despite those comments, product appeared in the wild in Germany for the Commodore Amiga, and rare copies when they appear on eBay command hundreds of Euros.

It was a tragic end to an era, and the sad end of an extraordinary company that created technically brilliant and wonderfully entertaining games for almost ten years. As per most British publishers, they were never able to take advantage of the American market, to compete with Infocom and secure the levels of sales that could have taken them well into the next decade. For the record, a few games did make it across the pond and did okay, notably *The Jewels of Darkness* trilogy published by Telecomsoft label Firebird.

[Mike] We were flattered by [the Infocom comparison]. We had huge respect for [them] – they created great games which we enjoyed playing. At the time, the two markets were very different. We didn't even have access to most of the popular platforms over there, for example the Commodore Pet in the early stages.

Britain's videogame industry, and particularly text adventure development suffered against the Europeans and the Americans because we had so many competing companies. The US text adventure market was dominated by Scott Adams and Adventure International at the start of the 80s and then matured with Infocom for the remaining years of the decade. They didn't have as many large, competing companies, or the vibrant indie or homebrew developers that were a stable ingredient of British gaming culture.

[Mike] I don't know. I've seen this in both my business since, the American home market is large and relatively affluent – disposable income is higher, so you've got a much bigger market and more people. Typically, companies can scale quickly if they become popular. With the UK, smaller markets, lower disposable income it was harder to scale so I think that's where small scale developers came from.

Though there is no data to loss of sales due to piracy in the 1980s, there is still doubt from anecdotal evidence that it was widespread. Cassettes were easy to duplicate, and high street store machines that made copying easier reduced legitimate sales at a time when profit margins were already thin.

[Pete] The main problem we had was with piracy. Something like 10 people played our games for each 1 that bought them. This meant that we could never make enough money for it to be viable and gave a huge advantage to the consoles that came to the market in the late 80s, and which were piracy-proof.

Even the introduction of Lenslok, an elaborate and controversial anti-piracy system that shipped a little plastic device that de-obfuscated a security code didn't fully stem the tide and recoup the costs of manufacturing and licensing the product.

[Mike] Did it multiply [sales] by 10? No, definitely not. I'd say probably the games we put in with Lenslok had maybe 50% more sales than ones without. So, it was substantial and significant, but it was also desperately inconvenient, you know, so very much a difficult decision to go for that.

The gaming media and press must also shoulder some blame for the demise for the genre. Text adventures continued to get better and more sophisticated throughout the 80s, and the quality of writing and puzzles improved as the parsing and supporting dictionaries made them more accessible for novice players. The magazines dined out on the sumptuous graphics of *The Pawn*, long before any arcade game was able to match the visual joys and capabilities of the 16-bit platforms. After that, they didn't take really feature and take advantage of further games from Scrolls or Level 9 games with just as spectacular visuals. The graphics in *Scapeghost* were just as good as those in early LucasFilm and Sierra point-and-click games that the press ultimately fawned over. The genre suffered a death knell as coverage dwindled and was shunted into an ever decreasingly small number of pages.

[Mike] Why things weren't selling? That's an interesting, and I think complex, complex thing that we probably never really will know. There was still a lot of copying games, and there was a lot of competition in the media about the consoles that were coming in. [...] The media started to ignore anything on a home micro, not just text adventures, which was frustrating because I felt a lot of publicity was given to imported games that were really expensive. [...] Public attention moved away for a few years and that was enough for studios like us to move on and do other stuff. We were doing other games in 1990 through to 1991.

Even with interest in 80s/90s games returning in the latter half of the 2000-decade, text adventures remained underrepresented and underappreciated culturally and underrepresented in videogame history. Whenever a documentary on games history appeared, there was very little coverage of adventure games. Even in Jason Scott's dedicated GET LAMP [**see Issue 06**] documentary, the final edit completely cut an interview with Richard Hewison. The British documentary From Bedrooms To Billions, a UK-centric film focused on the rise of the independent developer gave Level 9 short shrift.

[Mike] We did an interview going on for an hour, and it came out about 30 seconds or something. We aren't the main star, but it still did feel like text adventures were not even a footnote.

Level 9 obituaries will always make the unfair comparison between themselves and Infocom, and undeservedly never puts the former on the same level as the latter. American blogger Jimmy Maher commented "their catalogue is a hard sell to modern players in comparison with that of Infocom and even Magnetic Scrolls." For some games that was true, but Infocom and Magnetic Scrolls never tried to match the sophistication and efficiency of technology that Level 9 achieved either.

Up until the end they refused to cast aside their support for 8-bit computers, and for some baffling reason produced hamstrung cassette versions of games. Though *A-Code* was the ultimate language for portability, much effort, time and money must have been spent optimising text for the various versions, and the technical implementation of multi-format disks seemed a costly distraction too.

Perhaps abandoning the 8-bits earlier and concentrating solely on taking full advantages of the 16-bit machines the outcome may have been different, but I suspect the technical challenge was part of the attraction for the Austin trio. As a technical innovator their vision and achievement is unsurpassed, and as a great British adventure creator they left a legacy of some wonderful games.

DESERT ISLAND DUNGEONS

As their vessel is consigned to Davy Jones' locker, the Austin brothers, **Mike** and **Pete** row to safety upon our desert island with only five adventures to pass the time.

Any 3 Infocom games plus *Scapeghost* and *The Jewels of Darkness*.

I know that's a trilogy, but we make the rules here.

Format: Amstrad CPC/Plus (128K)
Publisher: Poly.Play Software
Developer: Doomsday Productions
Release Date: 2017
Website: www.doomsdayproductions.org

DOOMSDAY LOST ECHOES

The Classic Adventurer is blasted into space with **Alberto Riera**, who, alongside Daniel Castaño has created *Doomsday Lost Echoes* - a sumptuous graphic text adventure that extols the technical capabilities of the Amstrad CPC computer.

Born in 1980 in Gijón, north of Spain, Alberto Riera's love of computer games and programming started with the Amstrad CPC6128 - a machine popular in continental Spain, having been successfully exported alongside the ZX Spectrum computer.

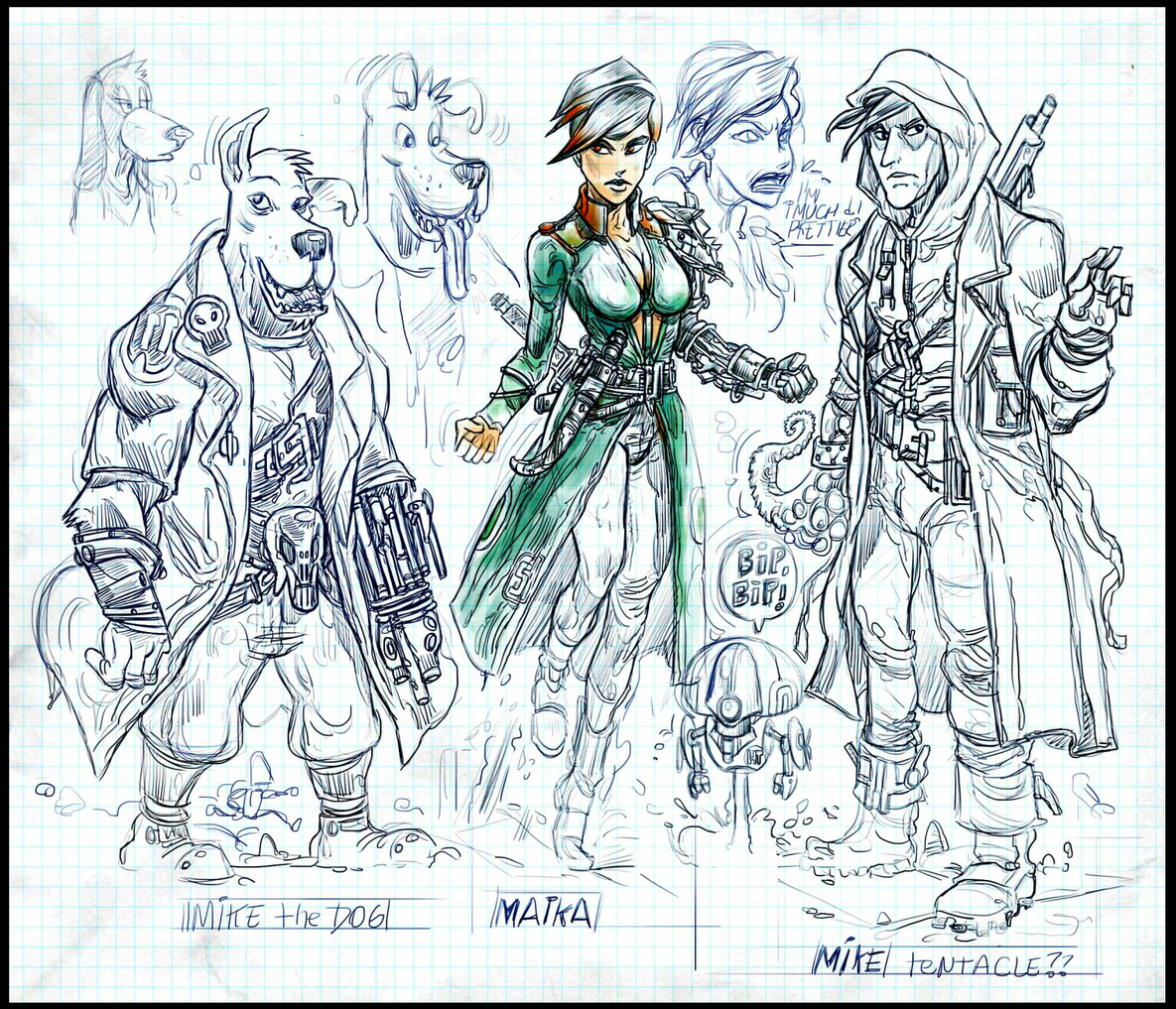
Spain nurtured a strong adventure market with companies like Aventuras AD [founded by Andrés Samudio after splitting from Dinamic] producing outstanding titles using Gilsoft's *The Quill* and *Professional Adventure Writing System [PAWS]*.

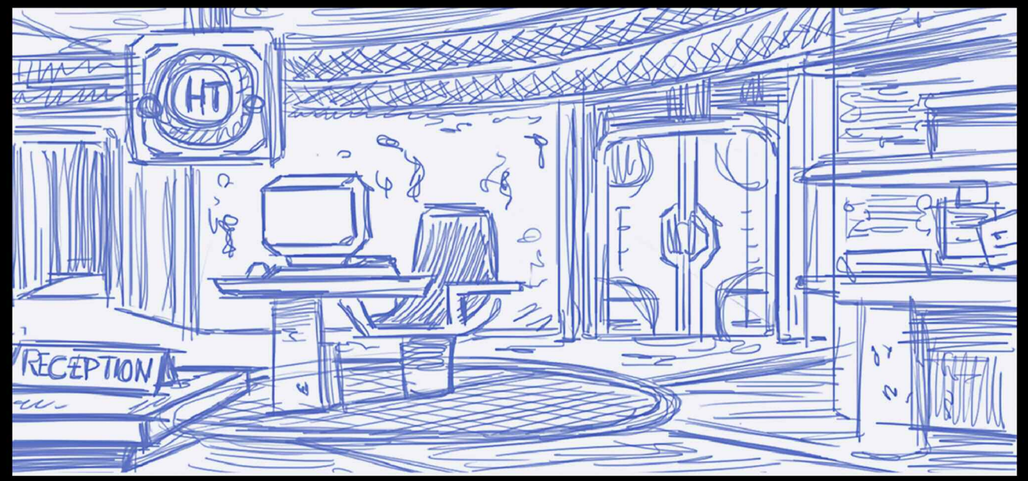
Tim Gilberts developed the *DAAD* for [Aventuras AD], a sort of super-*PAWS*, and they created a few wonderful titles using it: *Cozumel*, *Los Templos Sagrados*, *Chichen Itzá*, *La Aventura Original*, *Jabato* and *La Aventura Espacial*. Not many, but all memorable. Other studios published things as well: *La Guerra de las Vajillas*, *Don Quijote*, *Abracadabra* and *Megacorp* were all notable games.

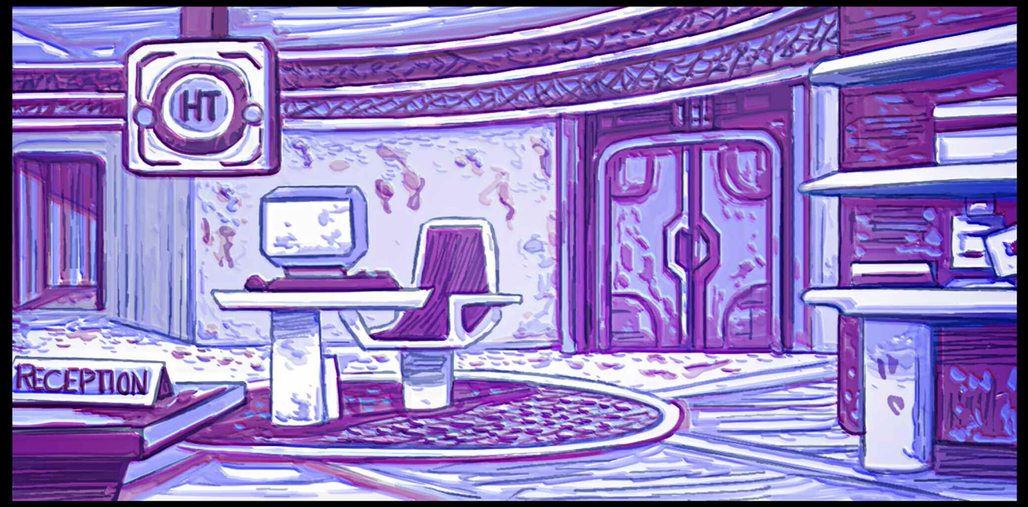
It was whilst chatting with a good friend, Daniel Castaño that the idea for *Doomsday Lost Echoes* [*DDLE*] emerged. They both wanted to get involved in the vibrant retro development scene and agreed that a text adventure would be an ideal vehicle.

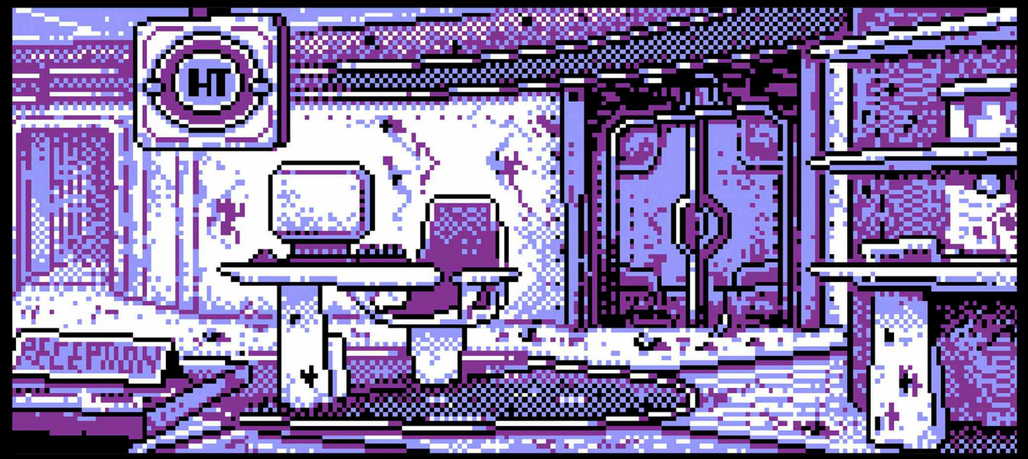
We wanted to contribute to the scene with our own little effort and, since I have always been in love with text adventures, it seemed the obvious choice for us.

Dani is a particularly kind person, with great talent and a brilliant mind. We have known each other for many years, way before we decided to develop something, and we use to have long conversations about videogames. Funnily enough, I came to know him because he is the brother of one of my best friend´s girlfriend and we were all living in the same city.









[Above] A small collection of concept artworks by graphic designer Daniel Castaño portraying the game's protagonists. **[Right]** The evolution of one of the game's early location graphics from sketch, to concept artwork, to pixelisation shows the commercial-grade production values invested into *Doomsday Lost Echoes*.

In *DDLE* you play the part of Mike, a renegade and bounty hunter who is offered a huge payday to track down a man called Arnold Craft. Craft was last seen over 30 years ago on the space station Regus, which for some reason now lies abandoned and aimlessly orbiting another planet. The adventure starts as Mike boards his craft, the Doomsday and makes the interstellar jump to the Pollux B system.

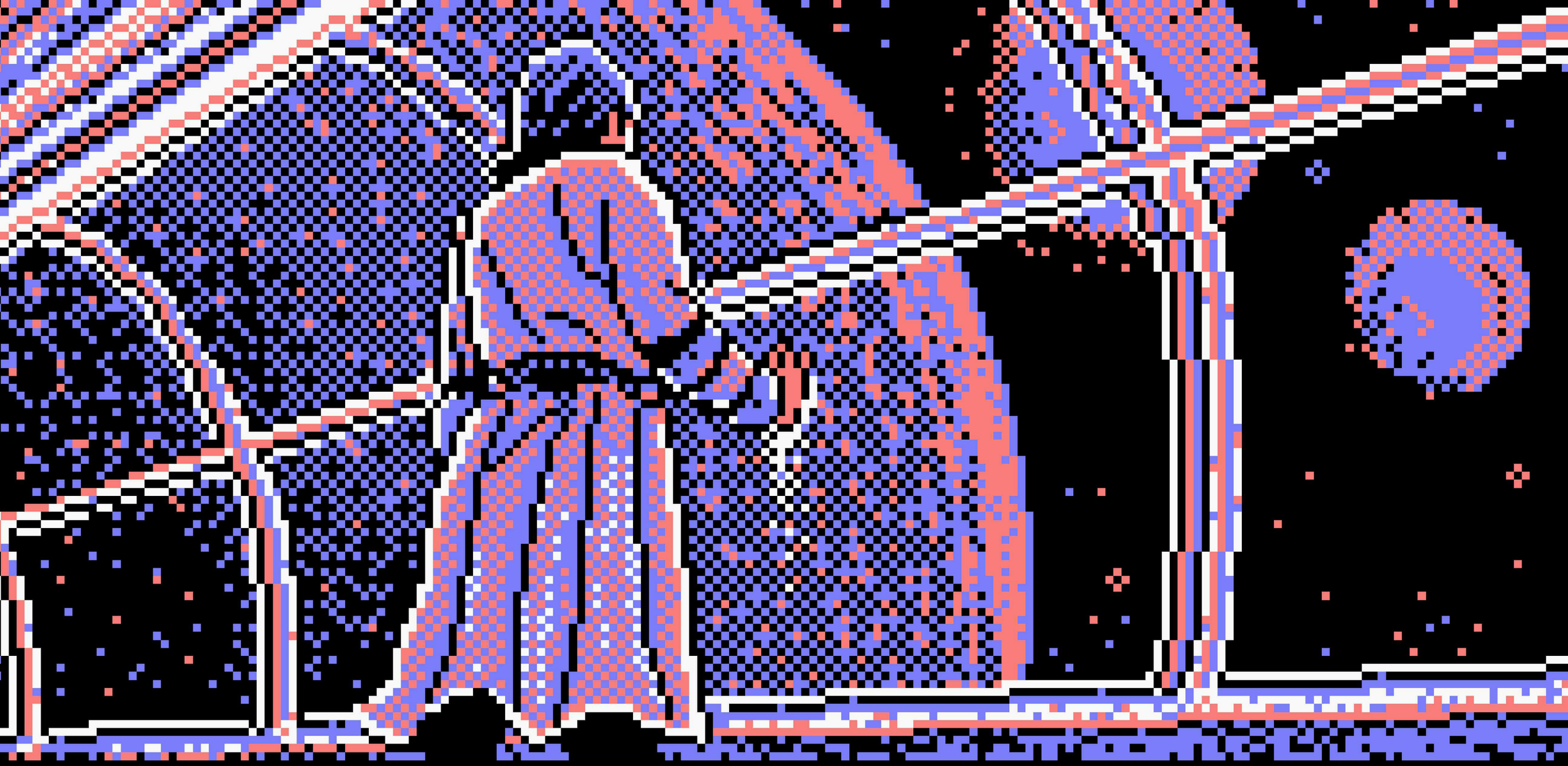
We first thought about the historical context of our game. When and where it would be happening, how the humanity would have evolved in that distant future. We wrote quite a few pages full of lore until everything fell into place. Then, we came up with Mental Mike, that can be considered our personal incarnation of Mad Max. We wanted him to be a very capable individual, able to survive in the most dangerous situations, and we wanted him to be very cynical as well.

From here, we moved on to the idea of the derelict space station and we chose a very clear and simple goal: find the missing person. Then it came the first draft of the plot, that was greatly expanded with new suggestions from [another friend] Themistocles Papassilekas. We ditched some ideas that we thought were not OK for a text adventure and we came up with quite a few others. Finally, we started working on the real game.

When Mike arrives at the Regus it quickly becomes clear that the station has been abandoned for a reason. As the plot unfolds, he realises that an infected crew member had brought a virus onboard and the resultant outbreak spread chaos and death throughout the station. It's not the most original of storylines, but it does hint at Alberto and Daniel's love of classic movies, games and pop culture.

We wanted our game to have a little bit of all those things that made the sci-fi so great in the 80s: big space ships, crazy monsters, and impossibly hard-boiled mercenaries.

We took a little bit from Alien, Mad Max, Predator, Commando and Total Recall. Regarding the games, *System Shock*, *Dead Space*, *Doom* and *Portal* had a clear influence, but many others were important as



well. We never tried to create something original, to be honest. The idea was to let the player play with something that was "new" but very familiar at the same time.

Unusually for a text adventure, three different conclusions to the story can be achieved. Each one offers a substantially different narrative and casts a contrasting light on events that have overtaken the space station.

We had the idea from the very beginning and I am glad we took the decision from the start. Otherwise, it would have been very difficult to adapt the code.

We decided to have three endings in order to offer some incentive to the most veteran adventurers and, at the same time, to produce a game appealing to newcomers. It is easy to just finish the adventure, but a bit more difficult to find the second ending. The third one is even more convoluted, so there is value in replaying it. In any case, we did not want to make a difficult adventure, but an enjoyable one. People do not have so much time these days.

Development began in 2015 and took a year and a half to complete, with the team working on the project during their spare time. A first beta was given to a band of trusted testers and the laborious task of bug fixing began.

We asked some veteran adventurers and newcomers to the genre for help. We chose seven testers and five of them found the commitment to finish the adventure and give us their feedback. All the bugs were written down and their suggestions taken into account. Some puzzles were made a bit easier, some others more difficult.

Getting that balance right, proved to be one of the more difficult parts of the process. Though *DDLE* does have some more complex puzzles, they are designed to be solved in a straight-forward and coherent manner.

There was a lot of discussion about them before they were implemented. We did not know if people would find puzzles too easy or too hard, and we wanted to keep all of them as logical as possible. At the end, the testers helped us a lot with this. There were a lot of people testing the game and many were veteran adventurers. Their feedback was essential.

We corrected all the bugs and we did some other polishing here and there. Finally, I checked the code around 30 times more - this is not a joke. Every time I checked there were new bugs, so I decided to go on and on until I was happy with the results.

[...] We decided to leave a few kilobytes of free RAM when the [game] was released just in case they were needed to correct bugs and things like that. We had ideas of how to use them, but we decided that it

IN SPACE NO ONE CAN READ YOUR REVIEW

After booting into a font selection screen [introduced to placate the Amstrad retro community] the story begins with a series of neat cutscene images that set the backdrop for Mike's predicament and his journey to the Regus. Once on the spaceship, he discovers an infected crewmate had brought aboard a deadly virus sending the base into chaos. That chaos has attracted the attention of several opportune aliens, and Mike spends the adventure scouring the station attempting to restore power, and wading through the horrors of dead crew members to complete his mission.

Throughout there are nods to retro gaming [mainly the Amstrad] and loads of pop and film culture references, including *Monkey Island*, The Creature from the Black Lagoon, Star Wars, *SpaceWar*, and predictably, Aliens. Every location has a graphic, all beautifully drawn, detailed and superbly coloured. At times you'd be mistaken for playing a Level 9 or Magnetic Scrolls ST title, the graphics are that good. Another subtle touch is the change in text colour to match the overall tone and hue of the displayed image. The text is lush, very descriptive, and the *PAWS*-powered parser has been made as forgiving as possible. It is friendly and intuitive, and recognises a wide range of verbs, nouns (even US-English spellings) and sentence structures.

There's no wonder development took so long. *Doomsday Lost Echoes* is a hugely ambitious and thoroughly professional game. It's gripping to play, and gives you an idea of what the 8-bit adventure market in Europe could have achieved with access to disk drive storage as a standard in homes.

would be better to be safe than sorry. Luckily, at the end, the extra RAM was used to add more lore, an easter egg.

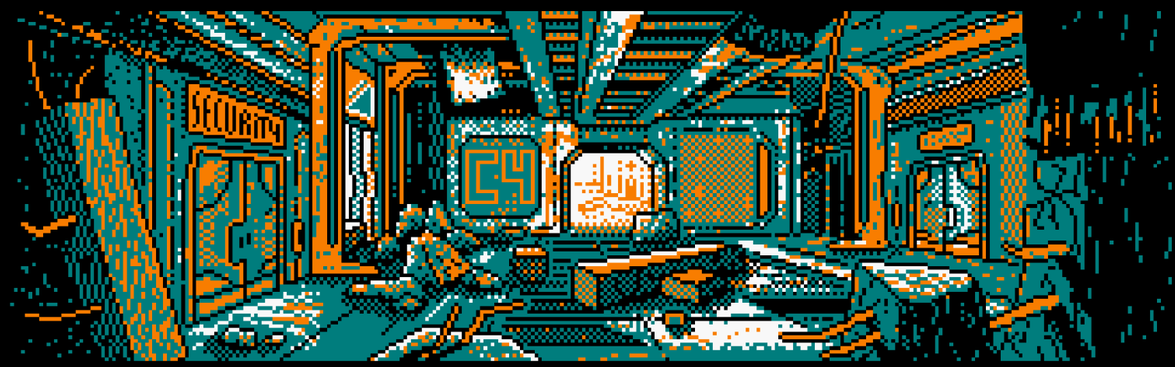
More on the Easter Egg later, but with Alberto satisfied, the first version of the game was released to the Amstrad community. It proved to be a double-edged sword and acted as a timely reminder of game development in the internet age - instant feedback and the visibility of conflicting opinions and tastes. In one example, a post complained that the game described events from a first person pronoun "I", rather than second person pronoun "you" in the text. A minor issue, but one that seemed to polarise attitudes.

This was totally intentional. We wanted Mike to have a very developed and clear personality and we wanted the player to feel that he was reading Mike's thoughts, but not impersonating the main character. I was never very fond of first-person approach in adventure games, I don't like the classical virtual narrator coming out of nowhere just to tell you what is happening. If there is a second part ever, the approach will be the same.

Its testament to the team that even under such pressure the original vision and ethos for the game was maintained, with minor changes, including the late addition of a font selection mechanism engineered into the final version.

I think that if you release a game for the community it is also logical to ask the community for feedback. Of course, you always should be loyal to your own vision of the things, but if there are improvements that are going to make the game better we always go for them. In this regard, [...] the adventure has improved [... thanks to the] members of the community.

As well as harvesting feedback from players, Alberto and Daniel became students of peer games and studied *Cozumel*, *La Aventura Original*, *Jabato*, and Magnetic Scrolls' *The Guild of Thieves* ("an




```
This is the central part of the hall.
The access to the stairs is here and it
would be possible to go to the lower
levels if it wasn't barricaded with
concrete blocks. I see the entrance to
the barracks to the east and the
security department doors to the west.
The security department doors are
closed.
>
```

[Above] Each location graphic has been meticulously drawn with oodles of detail. The font is well designed and legible and there's a subtle change of colour in each location to match the overall image pallette, adding a small, but visual touch of quality.

incredibly ambitious program") for their advanced programming techniques and implementations of different types of parser.

The resultant game is stunningly impressive and quite an accomplishment, making comprehensive use of Gilsoft's special edition of *PAWS* for the Amstrad CP/M. Making the task somewhat easier, the team could cross develop to the Amstrad using modern tools and IDEs.

[It] was actually easy. *PAWS* for the Amstrad CPC is just a compiler that runs in CP/M. You need to configure your emulator to be an Amstrad with two floppy drives, being the drive B a high density one with *PAWS* copied on it. CP/M is loaded from the drive A, then you switch to drive B and compile by copying the game code there and calling the compiler with the proper parameters. Of course, you want to speed things up because compiling can be very slow, so it is always good to increase the execution speed in the emulator as much as you can. This allows to compile and test the game in a matter of seconds or minutes.

The code was written using SciTE text editor, in Windows, and it is completely full of comments and annotations. Actually, it is almost 400KB. Luckily, the compiler ignores all this and just keeps the real commands. Being able to comment the program as much as I wanted was a blessing.

The CP/M version of *PAWS* builds games from text database files and didn't have any picture capability at all. The *CP/M Art Patch*, co-developed by Tim Gilberts and Graeme Yeandle in 1990 allowed the screen to be split and for pictures to be displayed in the upper section.

[The *PAWS* [...] native graphics patch allow[ed us] to use pixel art that is read from the floppy in real time, saving precious memory, and, since it is a standard CP/M program, you can fill the second RAM bank as you wish. This means that your code can be as big as 61KB. Including graphics, our adventure is almost 700KB, quite a monster for the CPC.

It was inspired by another Amstrad game using the same technology, called *NHeredia*, and techniques developed by Miguel Sky of ESP Soft. Using an additional piece of code, and amendments to the location database, pictures could then be read from files on the required attached disk drive.

The basic idea is simple and quite clever: you keep the pixel art in the floppy and an embedded database in the code that associates a location with a particular graphic. When the player moves, you scan the database and the new graphic is loaded directly into the video memory.

It integrates with the parser really well. I am not aware of commercial games using it back in the days, though, maybe because nobody wanted to develop for disk machines only. On the other hand, although developers could have added additional content as we did, that would have not been feasible from the commercial point of view, since you would be cutting your sales by more than half by not developing for 64KB machines. On top of that, our game uses a non-standard 3.5" floppy disk and almost nobody had one of those, something that would have cut the sales even further. In summary, it would have been possible, but a suicide.

Using the patch was relatively straight-forward, but it added an overhead and further complications to creating and compiling games. It didn't deter Albert and Daniel who started creating *DDLE's* incredible 60+ images using PC-based tools.

Dani had total freedom creating the material. I only provided him with the descriptions of the locations and the info regarding resolution and colour palette. He took care of the rest. All the

graphics were made in Photoshop, and as far as I know the process started by drawing them in black and white. Then the basic colours were defined and finally the patterns and dithering added by hand. The colours are assigned by the CPC in real time, so we had to play with them in some of the locations.

The ability to use bitmapped graphics for every location has its obvious benefit in quality. The images used in *DDLE* are superbly drawn, have a lovely use of the CPC pallette and are very, very detailed. So detailed in fact that they have been used to add to the playing experience providing visual clues to objects and puzzle elements.

We think that the graphics add a lot to the adventure and close observation is actually necessary to solve some of the puzzles. They also tell a story on their own. We had very limited RAM available so we wanted to use all the resources at hand to create the proper atmosphere and enrich the whole experience as much as possible.

At various junctures within the story, there are a range of cutscenes that play out as a series of individual images and page turning text.

"Centuries in the future humanity will realise that [the Amstrad] is the computer of choice to install in a space station stranded in the middle of nowhere"

It's another element that Alberto believes the experience doesn't distract players from the purity of a text based game.

I have always preferred adventures with graphics. I think that they can really contribute a lot to the whole experience and I believe that our game is no exception. It is a personal preference, though, many people choose to play text-only adventures and that is completely fine as well.

It's a smart use of graphics, and the care and attention gone into the *DDLE* visuals means it does stack up against many 16-bit titles. It gives a glimpse at what a disk-enabled version of *PAWS* could have been capable of. Unfortunately, disk drives wouldn't be commonplace until the arrival of the Atari ST and Amiga in much of Britain, and we are left wondering what UK writers and developers could have achieved with such a system. It is a shame the market never saw a general release of the equivalent engineered for Delta 4, the *System Without a Name* [*SWAN*].

The closest thing that exists is the *DAAD* for Amiga, Atari and PC. Some fantastic games were written with it from Aventuras AD I was mentioning before.

The *Diseñador de Aventuras AD* or *DAAD* [an acronym of AD Adventure Designer] for short, was an advanced version of PAWS developed by Tim Gilberts specifically for Aventuras AD. It was a tool that was only available commercially in Spain, but at a high cost.

It didn't handle 128K games as well as *PAWS*, so the resultant *DDLE* game code ruled out its use. It was also clear early on in development that attempting to create the game for the 64K Amstrad was out of the question.

[...] It was clear that it would not fit. Just for you to know, when the game is running only 300 bytes of RAM are still free.

Despite being limited to the elder tool [*PAWS*], it didn't hinder development in any way.

It would have been great if it had allowed us to use more complex code, for example loops and proper conditionals, [...] but it is what it is. Maybe I should have been cleverer when using it!

As for Alberto's recommendations to developing games with The Professional Adventure Writer?

Read the user manual a couple of times first. Then, when you have the general idea in mind, try to read all the old literature about the parser.

For example, the questions that people had back in the days when they were using it and the answers given to them. Everything is available to download and there is a treasure of info in these old documents. Besides that, I can only recommend to have your adventure completely written and sketched before start typing a single command.

Did *DDLE* adhere to any design rules?

Only a few. Keep the puzzles logical and fair, add rewards for the dedicated player, don't make your game artificially longer than necessary and avoid at all costs cheap deaths and frustrating situations. I don't really understand adventures where you can suddenly die without any kind of advice, or [those that are] filled with impossible puzzles. I think that we should try to leave all that behind.

Playing *DDLE* it's easy to forget that it has been created by a team who don't have English as a first language. The text is written to a consistent level throughout, with barely an error in grammar or spelling which must have proven difficult.

[...] I have been living in UK for more than eight years now and I am constantly writing and speaking in English. Still, English is not my mother tongue and, although the game was understandable, I felt that the language was very stiff and not natural. Therefore, three people with much more knowledge than me went through the texts.

What are your favourite elements to the game?

One of the things I like is the fact that the parser is not rigid compared with most 8-bit text adventures. We wanted to create a game that was playable by everybody. Therefore, there was a lot of thinking about how to parse the commands. If you lower the requirements too much people could be able to solve puzzles almost by chance, but if you are too picky frustration arises easily. I like the balance we managed to achieve. I also like the graphics a lot and the amount of lore we managed to squeeze in the adventure.

Would you go back and change anything?

Yes. If I had to code it again I would keep the intro as a separate program. This way, it could have been longer and we could have saved precious RAM. With more RAM we would have had space to implement a better inventory, one with actual pictures of the objects, and maybe extend some other elements, like the cut-scenes, or even add some sounds or little animations, like blinking lights.

Then there's the Easter Eggs. These seem to have been very important for the team, and there's a hint that clues exist in the game that break into the real world and cross over from the digital into the physical plain.

The main Easter Egg has not been found, but this is not a surprise, I would not be able to find it myself if I didn´t know about it. I can give you a hint, though. The video where Mike is shooting things is actually a puzzle, and solving it is essential to find the Easter Egg. We

PHYSICAL PERFECTION

Launching in 2002, Chronosoft were one of the first specialist publishers to revive distribution of games for retro consoles and computers. Since then, the popularity of physical media has grown, and many other producers have joined the retro scene offering high levels of quality and ever more sophisticated production techniques - some offering cassette, disk or even cartridge based games.

One of those, Poly.play, a German company, has secured the rights to retail Doomsday Lost Echoes. They have made two special Collector's Editions of *Doomsday* available for purchase.

One configuration can be bought with a 3" and 3.5" floppy disk and the other with dual 3.5" floppies, depends on your preferred formats.

Both have sumptuously designed artwork by Daniel Castaño and feature a wide range of extras, including a packed SD card, complete with digital wallpapers, icons, artwork, development notes and the emulator images for those who might just want to own the cardboard.

There's a well written and helpful 24-page manual that has a few hints and tips for the game, as well as an A3 poster and 4 different stickers.

Buy now from:

www.polyplay.xyz

promised a physical reward for whoever finds it and describes how he or she did the discovery. It is still sitting in my living room.

But, as mentioned before, the actual Easter Egg is only one of a few external references that appear in the game. It's obvious that the developers love the Amstrad, and the CPC makes plenty of appearances in the game, at one point being a computer still running on the abandoned space station.

Centuries in the future humanity will realise that [the Amstrad] is the computer of choice to install in a space station stranded in the middle of nowhere, particularly if it is orbiting a gas giant. They will be all Schneider, though, because they are shielded and have centronics ports. Edge connectors are not so reliable in extreme environments.

The game was finally finished in November 2016 and made available via the internet for free. Despite the complicated twin-disk emulation required, the website has a comprehensive and easy to follow setup guide, and also hosts an introduction to text adventure games for those unfamiliar with the genre. It seems to have helped the game gain traction and success so far.

We actually had more than one thousand downloads just in our own website, although there are quite a few other mirrors here and there. The physical edition is actually selling very well too, as far as I know. Moreover, many people have written e-mails telling us that the adventure is very enjoyable, and there have been many articles about it in lots of webs and magazines. It is very encouraging and we are extremely grateful to all the players and supporters.

Evidence of the game's quality and growing reputation was the acquisition of the publishing rights to a physical version by retro game publisher, Poly.play, headed by Sebastian Bach.

It is awesome that publishers like Poly.play exist, otherwise the physical edition would had never been a reality - and the same is true for many other games. We were directly approached by Sebastian, that offered us his services and took care of everything, including sourcing all the floppies and copying the game to them. We only provided all the material in digital format and he did the rest, it was a truly great experience.

It's a completely different experience having a physical product, and loading a game via cassette, or in this case a disk.

Yes, at least in my case having a physical release brings back many good childhood memories. The same is true when I play the game in my Amstrad, particularly if it is connected to a CRT monitor. The graphics were made with these screens in mind, same as in the old times, and they actually look much better displayed on them. The drawback playing in a real Amstrad is the speed. You can speed up the things in an emulator considerably. In any case, I think that the true experience beats emulation by far, particularly if you had an Amstrad back in the days.

Daniel designed all of the artwork and additional material required for the inside and outside of the box [see Physical Perfection box-out] as well as work on a user manual and other 'feelies' included. It has been exceptionally ambitious for a homebrew project.

We wanted to do something simple, but then it started to grow more and more... and at the end we were a bunch of people working very hard towards the same goal. Actually, I consider this adventure to be quite a community effort. It would have been impossible to release it without all the people helping us.

Will we ever see Mike make a return, perhaps with one of the other characters in the Doomsday after his escape from the Regus?

We are working, just Dani and me at the moment, on an action RPG. If we manage to finish it one day it will be a PC release. It is something much more ambitious and enormous compared with *Doomsday*, so we are not even sure about the feasibility of the project. Well, let's be honest, we suspect that we won't be able to finish it at all! We have worked quite a lot, but what we did is just the tip of the iceberg. In this new game I am, again, doing the coding and Dani is taking care of the graphics.

I must say, however, that we would love to make *Doomsday II* in the future if we had time. We even now how it would continue, and who would be the main character. I can tell you that Mike would not appear until the end of the game, that you would visit The Pit and the surface of a planet...

DESERT ISLAND DUNGEONS

After being stranded on space station Regus, and attempting to escape from Kepler 452b, *Doomsday Lost Echoes* coder **Alberto Riera** finds himself stuck on the text adventure desert island.

That is a difficult question, but I would definitely choose some of the games from Aventuras AD, and, as much as I love my Amstrad, I would go for the Amiga versions. The graphics were better, the parser more tolerant and the texts far more detailed.

I think I would choose ***La Aventura Original***, ***Jabato***, ***Cozumel***, ***Los Templos Sagrados*** and ***Chichen Itzá***.

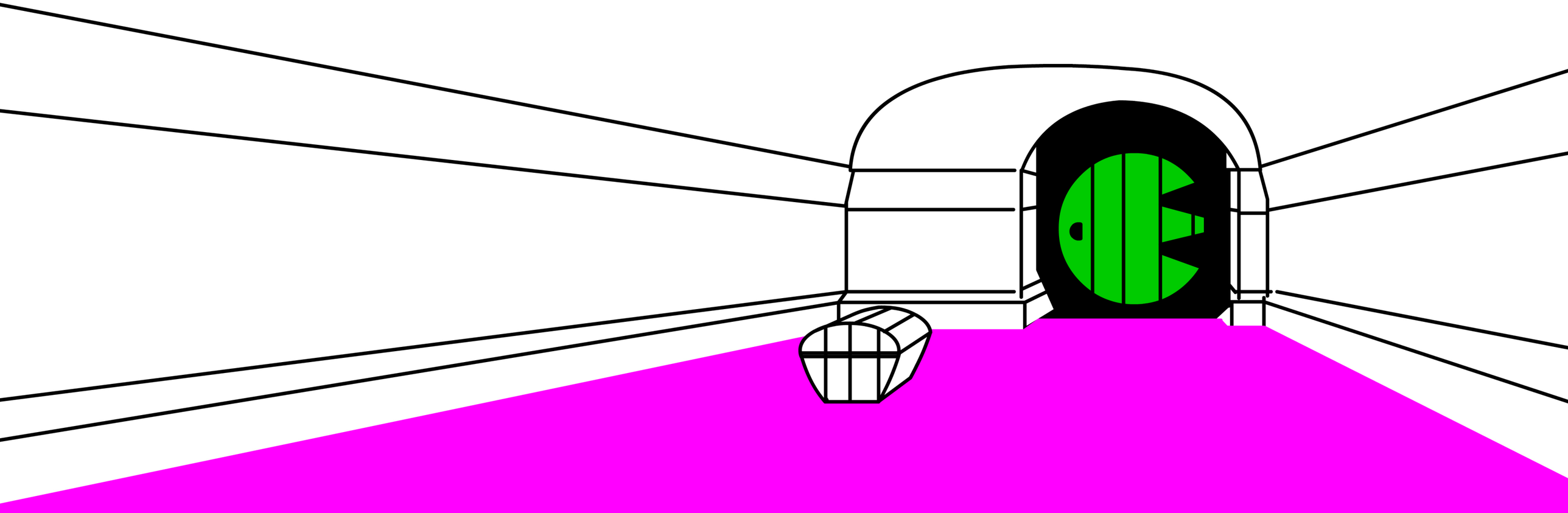
The first two are awesome adventures and the last three part of the same trilogy. I think that if I did not know the answers to all the puzzles, it would take me more than a year to finish them.

VERONIKA MEGLER

The Hobbit, created by **Veronika Megler** and Philip Mitchell captured the imagination of adventurers, entranced by its intelligent, autonomous characters, bold graphics and sophisticated parser. Almost four decades later, **Veronika** and **Alfred Milgrom** sit down and start singing about gold.

Alfred Milgrom founded Melbourne House as a general book publishing company in Britain in early 1978. Previously he'd been involved with an Australian company called Outback Press but because his office wasn't located in London, he fell afoul of licencing difficulties in obtaining rights for US titles to sell across the commonwealth.

[Alfred] Prior to book publishing [I] was at the University of Melbourne, where I completed a Bachelor of Science (Hons), and was working on my Ph.D. I had a strong background in science and had developed a strong interest in computers and programming. In February 1980 I read a newspaper article in the Australian Financial Review, about the start of game publishing in the US, with titles such



as *Chess* and adventure games by Adventure International, for example. I was excited because this field combined two strong interests of mine - computers and publishing.

The entrepreneur spent much of 1980 visiting the US, keeping one eye on the fledgling game publishing market that was fostered when the ZX80 was launched by Sinclair Research. It was the machine he'd been waiting for, and with an affordable retail price of £100 Alfred knew that it would be popular and positioned Melbourne House to take advantage of it.

[Alfred] Book publishing was what I knew, so in September 1980, I decided to write and publish the book '30 Programs for the ZX80'. This book gave the owners of the ZX80 an idea of what could be done with this computer, including a range of interesting techniques. Naomi Besen, who was my wife at that time, contacted Clive Sinclair to obtain an endorsement for this book. Clive Sinclair was very busy, and I think that he did not take the time to look at the book. However, after many repeated calls from Naomi he told his secretary to say that the book was excellent.

More books followed and sales soared. For a while Melbourne House experimented by importing Commodore PET and Apple II games from the US, but after little interest in a country with so few of the expensive machines, they ceased the operation. Not deterred by the poor sales, Alfred predicted that a home software market, buoyed by more of Sir Clive's cheap machines would soon become viable.

[Alfred] I realised that there was very little difference between developing material and putting that content onto paper or putting that content on a cassette tape. I very quickly started searching for someone to help develop content, and I employed a student to work during the university vacation.

Breaking from book publishing, Beam Software (named by taking the BE from Naomi Besen and AM from Alfred Milgrom) took tentative steps into the software development market and recruited its first student employee, William Tang. Tang produced *Strike Force* on the TRS80 and would go onto secure his own legacy by creating the iconic *Horace* series of games for British home computers, with Alfred contributing the design of the inimitable main character.

Whilst Tang was cloning *Pac-Man* for the first *Hungry Horace* game, Melbourne House, published its first titles for the ZX80, only to be caught unawares when Sinclair announced the availability of the ZX81. Before they could respond, in March 1982 the ZX Spectrum arrived making their existing product catalogue look tired overnight. "The demand for books for the ZX80 fell overnight" Alfred bemoaned to Home Computing Weekly about the obsolescence of his products, remarking that "[Sinclair] [then] did it again with the Spectrum!".

The speed of change with home micro technology was an important lesson for Milgrom; an early, stark warning of the short development lifecycles required to target new home computers, and the precarious nature of game sales that could suddenly fall off a cliff edge with the arrival of the next best thing.

Beam continued to publish books and arcade titles, but to put his team on the map, Milgrom wanted to develop a text adventure. He'd been a part of distributing Scott Adams' successful derivatives of *Adventure* and seen early adaptations from Artic and Abersoft reaching the British market. He wanted to create something bigger, better and more inventive than anything currently available. The design had to be encapsulate future and forward thinking, and any code needed the adaptability to take advantage of any new systems that would arise during development. But first, they needed a subject.

[Alfred] It was my intention to develop adventure games, especially to go beyond the simple two-word user interface other people had used in such games, and I wanted to have a rich and emotive environment and story. The obvious starting point for fantasy seemed to me to be the Tolkien books, and especially The Hobbit.

With Tang employed writing other books and expanding the *Horace* series, Alfred needed to supplement his workforce to undertake the new project. It had been a favourable experience employing a student, so he adopted the same approach and posted an advert seeking programmers on a notice board back at Melbourne University.

A young female Computer Science student called Veronika Megler responded to the advert, looking for a more flexible part-time job than her current role as a computer operator.

[Veronika] I think it just said it was a programming job – the ad was just a few words long and a phone number, I remember. I didn't have to code to get the job. I did the initial design within the first few hours and reviewed it with him, as I recall.

In that startup year, Beam operated out of Alfred Milgrom's living room in Melbourne. As they expanded, they moved into Suite 4; a large, open plan office situated at 75 Palmerston Crescent in the south of the city. Megler arrived to be greeted with a large, vacated space and a series of haphazard empty tables.

[Veronika] [...] Over time they started getting more and more computers; initially Dick Smith TRS-80 knockoffs, then later Spectrum and other systems appeared. My colleagues from Uni, Keryn and Ray, joined us, and wrote short BASIC games for [Melbourne House's] published books. Keryn in particular was my usual partner in pair projects at Uni. At Melbourne House she would sit chuckling as she wrote some new game. It's a different style and skill from what I did.

Veronika had the perfect aptitude for the task ahead, understood the challenge and showed talent and immense potential. She was hired on the spot. Alfred gave her a single specific instruction: "Write the best adventure game ever. Period."

The reference game for all early efforts, was Crowther and Woods' *Adventure* and most home computer versions on the market had set out to emulate the original classic. To Megler, it seemed sterile and repetitive. In her Case History Of Writing *The Hobbit* she commented "once you'd figured out the map and solved the puzzles, it was instantly boring. It played the same way every time."

The non-playable characters [NPCs] found in the caves came in for specific criticism. Though the locations would remain static on any subsequent plays of the game, there was an expectation that the creatures and beings that inhabited the world wouldn't – they'd move around, respond in different ways to the presence of the player and have some ability to interact with each other and their surroundings.

Secondly, the list of typed actions that the original *Adventure* recognised was limited. The frustration of hunting for the correct verb and noun to progress in the game vexed Megler. "You had to figure out exactly the incantation the game expected; if the game expected KILL TROLL, then any other command - ATTACK THE TROLL, for example–would get an error message. You could spend a long time trying to figure out what command the game developer intended you to issue; as a result, most adventure games tended to have the same actions, paired with the same vocabulary".

So, there were two distinct areas to improve, and Veronika recognised that her skillset was better suited to addressing the game's logic. To compliment her, she needed was a colleague to work on the game's parser, and she had the ideal candidate in mind, a fellow Computer Sciences student who had partnered with her on group projects, Philip Mitchell.

[Veronika] Phil was a much more algorithmic, semantic and correct programmer than I was, or am, more of a pure software developer. Perhaps more of a convergent thinker. [...] I'm a divergent thinker – very creative, excellent at integrating ideas from many places and disciplines in unusual ways to solve "white space" problems, but not nearly so rigorous or precise. [...] I'll come up with a solution that works to some level of accuracy. But the repetition of refinement, streamlining, successive improvement bores me very quickly – it's a different kind of thinking. Phil was much more precise, correct, algorithmic in his approach, and optimizing algorithms was his jam. He was also excellent at optimizing memory usage and dealing directly with hardware.

She introduced Philip to Alfred, and he was duly hired to work alongside her. He was an obvious choice, and Mitchell himself felt that his knowledge of hardware was an enormous contributing factor to him being employed. "Being able to understand at a level

the way a micro is doing things is easier to get to grips with a new machine" he told Computer Answers in 1984.

With Michell onboard, it was one of the first times that a manufactured selection of people had been brought together with the purpose of home videogame development. In 1982, that concept of a team of individuals each with a particular skillset required to craft a game was unheard of, and revolutionary.

[Veronika] I've never considered that, but yes, that's true. We really treated it more as a software development project than as a hobby, which was very unusual for the time. [...] At Uni, for pair projects, I always worked with Keryn, and Phil always worked with Ray. [...] We were all pretty serious nerds and hung out with the other series nerds in the Computer Science building. We'd also have regular evenings in a games arcade. Alfred would give us $20 to see what new games were doing. We'd watch one person playing while the others were analysing the game. But I also had a separate group of non-computer science friends that I partied with.

The team knew they required a story, but neither member could come up with something vaguely compelling. They toyed with the idea of bringing in a writer, but was going to be a cost that couldn't be met. The next best thing was to settle on working from an adventure that was already written. After Alfred's suggestion, they all agreed they enjoyed Tolkien, and even though Lord of the Rings was considered, it was a universal decision between the three protagonists that The Hobbit would offer a more compact narrative for a single adventure.

[Veronika] [Alfred] suggested The Hobbit. It seemed like a good idea at the time. We didn't consider the licensing beforehand. Fred took on the licensing discussion and kept Phil and me out of it – thankfully.

Through Melbourne House's existing relationships, Milgrom was able to approach the publishers of the books and start negotiations directly with the Tolkien Estate and the Saul Zaentz Company for the rights. Licencing The Hobbit was straight-forward and clear-cut, unlike the Lords of the Rings [**see Special Edition**], but the risk of not securing a deal left the danger of a potential IP banana skin for the future. As an insurance plan, Megler and Mitchell embarked on the construction of a generic fantasy game, something that could easily be adapted to a range of themes and narratives - just in case the official endorsement of Middle Earth slipped through their fingers.

[Veronika] I designed it with the specific intent to be a reusable games engine. My idea was that you'd be able to replace the "databases", as I thought of them – the map, the list of NPCs and their actions – and have a completely different game. I thought that way you'd be able to turn out a series of games quite quickly using the same engines. [...] The structure of the "animals" database and the character action sequence was defined within my first few hours of game design and never changed. [...] What I didn't take into account was that I'd start adding more complex interactions between NPCs and players than the simple, stilted fights from *Adventure*, and more complex puzzles. Then all the special cases required so much additional work. In that I was still thinking a little more generically and simply. [...] [I then] layered onto it more sophisticated character action functions, to support some of the more interactive puzzles. But even that concept, that each "action" consisted of a verb, a function, and an object, which could be null, was inherent from the beginning.

They split the project's workload down the middle, Veronika developing the game's interpreter, logic and database structure, and Philip the interface and natural language parser. Before she could get started, there was the boggy swamp of source material to wade through.

[Veronika] I'd read the entire Tolkien series several times long before I started writing the game and was a big fan of it. The Hobbit seemed the only one out of the series that I thought would be possible to convert, as the other books have far too much richness and complexity. [...] I saw the game as an abstraction of the book, in the same way that a movie simplifies and hints at a book. It was an

interesting challenge to decide what to keep and what to leave out, what I could do within the technical limitations, and how to still reflect the book. I tried hard to capture the essence of the key places, characters and events. I must have done a good job of it; many people have complained about being annoyed by some of the characters' behaviour, but I've never heard any complaints that the game misrepresented the book.

While the epic tale was being manipulated into more manageable chunks, Mitchell made a start on the parser. There's no hiding this was Melbourne House's focus of ambition for the game from the outset. Milgrom's aspiration weighed heavily on the construction of a more elaborate way for players to interact with the game's world. He wanted to go "beyond the simple two-word user interface" he explained to Computer Answers magazine, outlining his vision by saying "we were looking for a much more sophisticated language analysis and dictionary including as many words as possible".

Stuart Ritchie [another student undertaking a dual degree in English Linguistics and Computer Science] was drafted in to provide another niche competency. It seemed from the outset that Ritchie was fundamentally a consultant, someone to provide guidance and academic language expertise – but his overall contribution is unclear. He certainly didn't provide any programming input, and according to discussions between Megler and Mitchell very little of what he furnished made it into the final game.

[Veronika] Stuart did quite a bit of work on the theory, but from what Phil told me privately, it wasn't in a form that was useful for coding [the parser]. I think Phil used it as education in language theory, and as an inspiration and a starting point, but then he developed the code based on his own design. In essence, we needed two components: something that could take complex sentences and convert them into "[qualifier] verb object" for the game engine to operate on; and then a way to take the results of the action and convert that into a set of sentences. All of that was developed by Phil. So, I'd position Stuart as a consultant and advisor to Phil, but Phil as the inventor.

Nevertheless, *The Hobbit* accompanying manual immortalised Ritchie in the game's credits: "This program structure was developed as a group effort by Philip Mitchell and Veronika Megler with Alfred Milgrom and Stuart Ritchie over a period of eighteen months." For his part, and is own legacy, he almost certainly influenced the semantics and thinking behind *The Hobbit's* brand-new parser technology, now named *INGLISH*.

The Hobbit user guide described the new mechanics as thus: "The rules of *INGLISH* are simple. The main thing to keep in mind is that each instruction must be in the form of verb-then-noun, where the noun or pronoun can also be implied." In practice it meant that no matter what the user entered, the parser would split the verb, noun and pronoun and hand it to the interpreter to undertake the appropriate action. TAKE THE SHARP SWORD AND EXCITEDLY HACK AT THE EVIL TROLL would be broken down and converted to a simple action KILL TROLL WITH SWORD. Megler explained in her case history that compound sentences would also be handled and parsed as a series of actions. TAKE THE HAMMER AND HIT GANDALF WITH IT would become two individual actions, GET HAMMER followed by HIT GANDALF WITH HAMMER.

Learning from the ill-timed release of their ZX81 and TRS80 games, Milgrom assessed the popularity of new home micros coming onto the market as potential launch machines for *The Hobbit*. The original

"fantasy story" adventure program, as well as being designed to be reusable, was engineered to be machine and implementation language independent. Whatever computer was around at launch time, the game could be easily compiled for it. It was a game *engine* 20 years before the phrase was coined, and like Scott Adams, Ken Reed and the Austin Brothers at Level 9 Computing, it seemed that adventure engineers were ahead of their time with such thinking.

[Veronika] I did no research and didn't look at the [*Colossal Cave*] code – such is the arrogance of youth! *Classic Adventure* was the only adventure I knew of or had played. No other adventure games came my way, and of course the internet wasn't around then to tell me of any others. I put together the overall design for a game that would remove the non-language limitations within a couple of hours on my first day on the job.

The Hobbit interpreter contained three database abstractions: locations, objects and characters. The first was straight-forward, a collection of game locations, information on which directions the player can take from each, and to which subsequent location the game then moves to. "Overrides" were added – rules that could be applied that had to be met before the player could move. In the best tradition of adventures, a randomly generated exit rule was implemented, and used to perfection in *The Hobbit* with the creation of the Misty Mountains' two connected multiple-location mazes.

[Veronika] Dynamic mazes just seemed to me a logical outcome of the confusion of mazes, and how they lost interest once they were mapped and became known. How can you recreate that interest? By having the maze shift, and then you get back that feeling of wait, I thought I knew this, how can that turn not be there when it's clearly in my notes? But the others were not a fan and actively discouraged me from going there.

The magic of The Hobbit is inherited from this style of thinking from Veronika. Megler's vision of what the game's objects and characters *could* become were laid as the bedrock for the game's posterity. A simple tweak to the objects database made them more complex by the addition of characteristics; weight, size, opacity and whether they could act as a light source or as a container and hold other objects. Those attributes could be extended to the characters, and it's this ground-breaking, and most significant development moment in *The Hobbit*, taking the immobile and passive monsters of *Adventure* and handing them the ability to execute seemingly intelligent and independent behaviours. Megler gave everything life.

[Veronika] The characters in *Adventure* bored me. They were so static and limited. I came up with an implementation that made sense to me. I quickly discovered that combining several simple innovations can cause the overall system to exhibit quite complex behaviour, and I loved that.

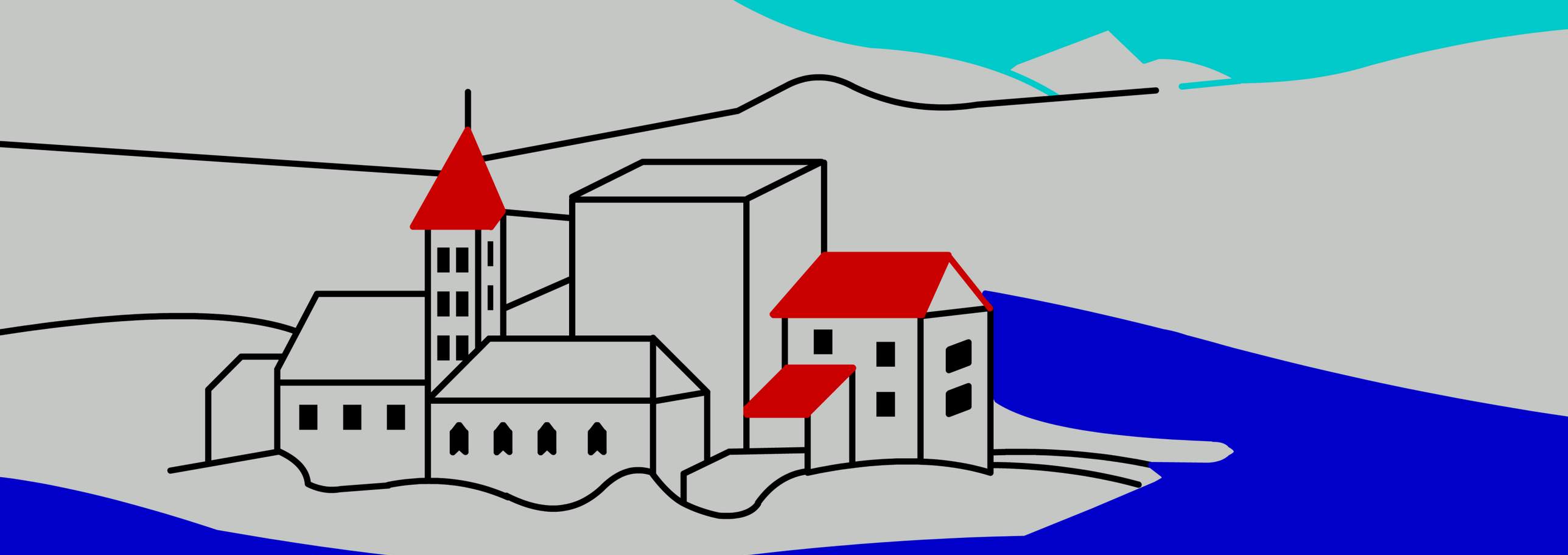
Her implementation gifted each character a set of actions that they *may* perform, a sequence in which they were generally performed and a frequency of repetition. The actions were in the main, drawn from the commands that the player could do – movement, object manipulation and the ability to fight with other characters. Again, "overrides" were added, so characters could be tailored to have individual traits and characteristics – and given a sense of personality.

[Veronika] It involved quite a bit of trial and error. Initially they were all a little too high on aggression, and it was too common to walk into a location and discover most of the NPCs already dead and then the rest to attack the player. But I toned them down a bit and liked the result.

Early previews suggested that characters possessed a rudimentary memory and if they received kind or aggressive behaviour from the player, they would remember in future encounters. Certainly, characters who did retain conversations they experienced played a part in *Sherlock* [**see later and Issue 08**], a game that would later use an iteration of *The Hobbit* engine to great effect.

[Veronika] They didn't have "memory", per se. What they did have were alternate characters, which could replace the current character. In essence, an NPC's character was a sequence of actions; or occasionally, some action function would set a flag that could be used by a later action function to allow, refuse or modify a response. For example, Gandalf's character included sequences something like: PICK UP OBJECT, since he was known to be a kleptomaniac; GO IN DIRECTION; SAY SENTENCE; and DROP OBJECT. At character initialisation, a random starting point in the action sequence was chosen for each character, giving the game some of its non-determinism. There was a "friendly dwarf" with his own action sequence, dominated by friendly or neutral actions. Then, if you attacked the dwarf, that would cause the dwarf's description to be updated to "angry dwarf", and the pointer to his action sequence to be pointed to a different action list, dominated by "attack" actions.

Even through it was just an application of these basic attribute flags, it gave the impression to players that NPCs were making dynamic and independent choices. It also meant that behaviour could be aligned with the characters who appeared in Middle Earth, albeit at a basic level. Trolls and Goblins wandered around wanting to kill everything, whereas accomplices of Bilbo were more troublesome. Thorin Oakenshield for example, tended to follow the player and without interaction became disinterested in the quest. Veronika pondered on the bored Dwarf's thoughts, and taking her inspiration from Tolkien's book implemented an action that has become a legendary memory of the game. With nothing better to do, Thorin, bless him, would sit down and start singing about gold.



[Veronika] Yes, I did write the now classic phrase. [...] I hereby apologise for how frequently he said that. Short character action list, you see.

Picking character idiosyncrasies that were "quintessential" in the book extended to other characters. Bilbo throughout the tale required the assistance of his accomplices at many junctures, and to allow the player to tap into NPCs actions Mitchell modified *INGLISH* to include a SAY TO command. It became a vital puzzle element, and NPCs could be asked to perform tasks or undertake commands on their player's behalf. Getting help wasn't guaranteed though, and NPCs often refused to do what was suggested.

Melbourne House dubbed this *ANIMTALK*, and the character interactions *ANIMACTION*. The final stroke of genius came as Megler and Mitchell realised that everything in the game waited for the player to type in a command. What they needed was a way for the world to operate on its own, and for the characters to continue to move and make their decisions even if the player had decided not to do anything.

[Veronika] It seemed an obvious extension of the NPC concept, to me. Once you had the other characters running around the game and playing it themselves, even when you weren't "in the room", so to speak – it just seemed wrong to have them all wait while the "player" went off to brew another cup of tea. But as with so many innovations, it started out as a joke. It was a very small code addition for Phil to make. In retrospect it was another step that made the game seem even more self-generative, rather than something controlled by the player.

But despite these variables, multiple plays of the game still led to some predictable results during testing. An extra dimension of flexibility was added. Instead of a linear iteration through the pre-determined action lists, Megler allowed characters to "branch to a different part of the sequence and continue from there or even jump to a random location in the sequence". It became a game changer, multiplying a small number of set actions into a behaviour list with literally thousands of variants.

[Veronika] The division between inanimate object and NPC was [also] left intentionally a little blurry, giving extra flexibility. For example, the object overrides could also be used to modify character behaviour. I actually coded an override where, if the player typed "turn on the angry dwarf", he turned into a "randy dwarf" and followed the player around propositioning him. If he was later turned off, he'd return to being the angry dwarf and start trying to kill any live character. Fred and Phil made me take that routine out.

It was the result that Veronika craved - no two games from that moment would ever be alike. But it came at a cost. The huge set of variables meant that it was almost impossible to test, as she explained in her case history, "small changes in starting conditions, [for example] initial game settings all generated by the random number generator, would lead to large differences in how the game proceeded.The earlier capability of balancing NPC behaviour was now lost. You couldn't predict where the player would encounter another character, let alone determine what actions they would take and in which order."

In some cases, it left the adventure in a state where it couldn't be completed. This seemingly anarchic ruleset is an endearing feature of *The Hobbit* and quite within the philosophy of its developer. Whereas the "randy dwarf" was tempered, Veronika's attitude to this chaos was to embrace it. She accepted that the game allowed people to do different things, some things that had not occurred to her, and in doing so handed the game its soul and breathed life into the characters.

[Veronika] Absolutely. I was really aiming for something like life, where the outcome is the result of many independent occurrences and decisions by many people, and sometimes things just don't work out. For me it was never a question – I actively wanted the unpredictability. Phil worked long and hard to get a true random number generator rather than the pseudo-random generators available, so that when I wanted randomness I could guarantee the game wouldn't play the same each time. In retrospect I'm amazed that the others were willing to accept it. I must have made some compelling arguments!

As development progressed Milgrom would pay regular visits to the team. Veronika recalls that they were a fairly self-motivated bunch, as the goal for the game was clear, so he trusted them to work independently when he wasn't around.

[Veronika] we also worked off-hours quite a bit, evenings and weekends, where we wouldn't expect him to be there and weren't really aware of where he was. I don't recall us having phone conferences with London. I never felt impacted by his absences.

When he was there, he'd take a cursory glance over Philip's and Veronika's shoulders whilst they were testing and was remembered

to exclaim "look at this!" when showed one of the first playable versions. Six months in and with a working copy in hand, Milgrom now pursued the book's licence holders with extra vigour. The demo was ready to dazzle, complete with a primitive implementation of location graphics.

[Veronika] The graphics came in relatively early [and] were already included. It didn't impact me at all, and I don't recall Phil doing any redesign. We liked the idea of a "graphic adventure game", and this was the reasonable step in that direction. We'd integrated the two game halves [and] I believe Phil compressed the images so they didn't take too much space. I do remember making him add the feature to turn them off, because waiting for them to draw was driving me crazy and slowing down my testing. There was still a lot of testing and debugging to do; there were many individual "actions", particularly with the way the NPCs and player could interact, that could cause crashes. While the coding was doable, thorough testing of a game of that complexity was beyond the capabilities of the tools available at the time. That piece we had not thought through sufficiently.

After the duo developed *Penetrator*, Melbourne House's answer to Konami's shoot-em up *Scramble*, Mitchell had a better understanding of how to optimise the graphics routines used for the location drawings. Mitchell rendered some rough imagery into the game using a minimal amount of precious memory and a highly efficient [if someone laborious] plot, draw and fill routine.

[Veronika] We did end up including fewer pictures than we might otherwise have, and if I remember right they were less detailed than the original drawings. Phil also spent time compressing them and optimizing the generation – far beyond what was the norm in games of the time.

The Tolkien Trust, and George Allen and Unwin Publishers were impressed, but negotiating the deal became a stumbling block. This was a videogame market in its infancy and many traditional media companies didn't know what a computer game was, yet alone appreciate any potential level of sales. Because of that naivety, the Trust agreed to grant Beam a licence on the sole condition that it packaged a copy of the book with each game.

[Alfred] They would obviously make money from the sale of the books, hopefully expand the audience for the works of Tolkien and keep the Tolkien legacy going in its original form. This was something they could understand and they very kindly accepted our offer.

It is worth stressing that they were gambling, and the licence wasn't secured until well into the game's development. Megler confirmed as much in her 2022 interview with Graeme Mason for The Guardian stating that she already designed much of the game's engine, ambivalent to whether Bilbo Baggins would be the central character or not.

[Veronika] I was building a generic adventure game. It wasn't until we had the underpinnings and reached the point where we really needed a storyline to go further that we discussed what the story should be. We were all Tolkien fans, and The Hobbit seemed the most accessible, certainly for a first game. [...] The only change I can recall was when I was adding puzzles or actions that required the NPCs to cooperate with the player. But I suspect I would have ended up there with almost any story. It was almost a logical next step from having NPCs in the first place.

At the end of 1981, with the project virtually complete and entering the final stages of polish, Veronika considered her future and somehow, "games" didn't seem like a viable career.

[Veronika] Games was seen as a hobby market, and not as "real" job. IBM, and other potential employers, HP and Digital Equipment made it very clear that having written a game was not regarded as any kind of relevant work experience and did not give me any credit for it. Since they had so much more experience than I did in the business and computer world, I accepted that judgment.

Despite pleas and offers of a full time role at Melbourne House from Milgrom, Megler took a full-time job with IBM at the beginning of 1982 and lost touch with her game and the games industry completely.

[Veronika] I think [Alfred] raised it, but I wasn't interested. I regarded Melbourne House as a temporary part-time job, and there was no real games industry at that time, it wasn't considered a "real" industry. There was no career path, just more of the same. I was ready for something new and different. [...] I never regretted moving on. It was a set of ideas, a time; and I implemented them and saw them become reality. After that it would have been iteration, refinement, more of the same.

As well as the professional factors, there was also the more personal matter of fatigue. Having worked long hours, weekends and being asked to code arcade game *Penetrator* as a side project, she was exhausted and feeling the detrimental effects of what became an industry staple – churn and burn-out.

[Veronika] Oh, I was definitely tired of testing and debugging Assembler by that time. From my perspective, the interesting parts had long been completed and the prospect of continuing to do the same thing over and over again held no real appeal for me. Had I chosen to stay at Melbourne House, there was no promotion path, just doing more of the same. Of course, in retrospect, it's had more impact than anything else I've done in my career – but I had no way of knowing that at the time.

With his development partner clearing her desk, Mitchell was handed the sole task of finishing the game and worked to improve the line and fill graphics engine he'd started earlier. Beam commissioned artist Kent Rees to assist, and he painted around 30 locations [most of which Mitchell had to simplify to meet the graphics capabilities of his routines] for the final product , including perhaps the most recognisable picture in a text adventure – a comfortable tunnel-like hall.

Picking up someone else's code could have proved to be a difficult proposition, but thankfully Veronika's diligence and professionalism meant that Philip was left with well documented and descriptive assembler source.

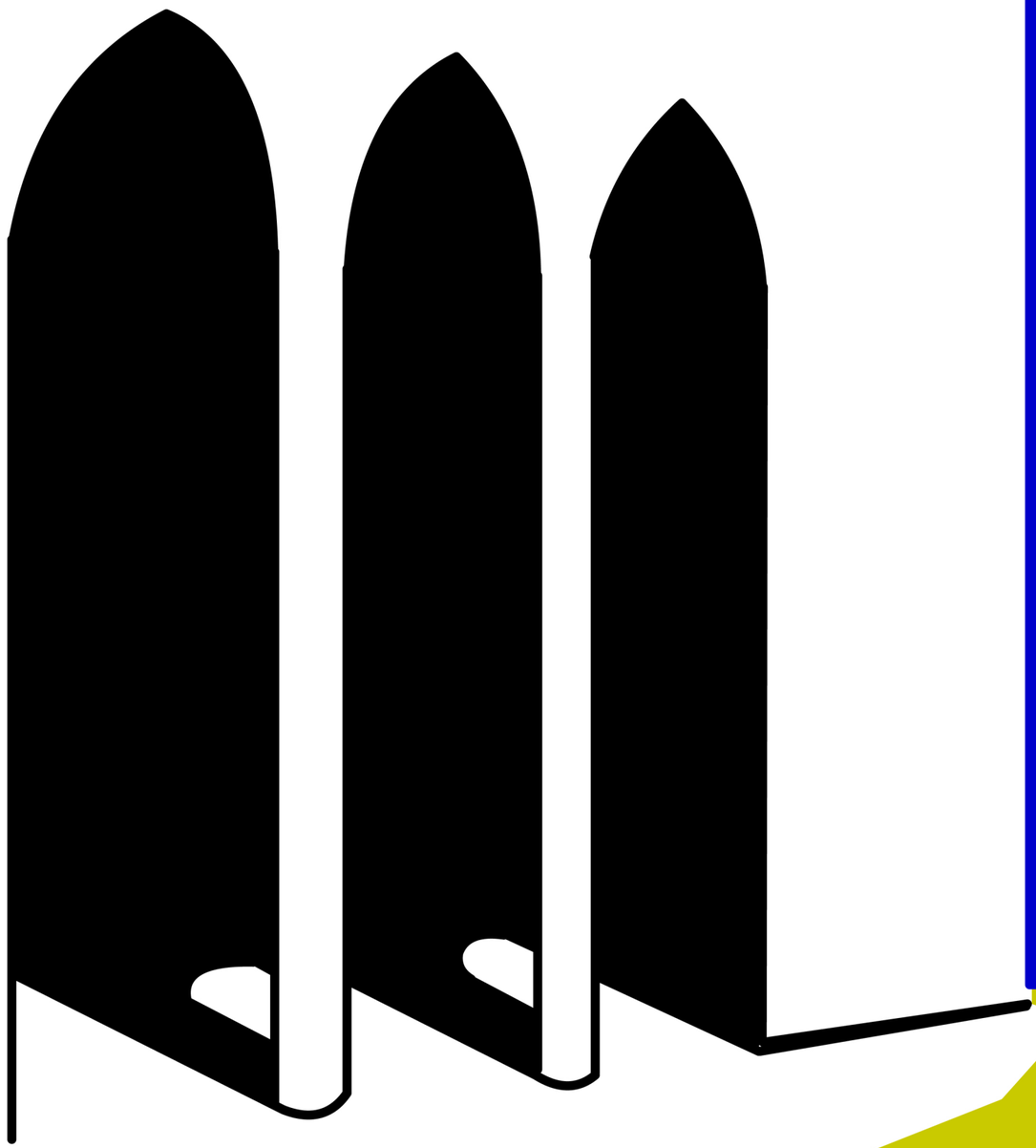
[Veronika] [...] [Phil] might say different, [but] even in my PhD: I spent hours once debugging a problem to find the issue was the initial setting of a variable. And the comment line said, "I have no idea what initial value to use here, so setting it to 1 for now". It completely clarified for me why it was set the way it was, and by then I knew how that value should be set. Without the comment, I'd have torn out my hair trying to understand an arbitrary decision meant to be temporary. [...] From my perspective, when I left the game was complete but was undergoing final tested and debugging. But the tester was a good tester and found problems, and of course debugging is a huge effort, especially when a non-deterministic game is written in Assembler. And the game allowed you to do so many things that it never occurred to us people would try and do. So, it's probably accurate to say that happy/expected paths worked, but there were just so many corner cases. [...] As far as I know the game itself didn't change in any material ways between those times. The code was patched for errors found.

The Hobbit was released in December 1982. It predated the big glossy full-colour home computer magazines of the following year and made a rather low-key appearance in black and white classified adverts. For a game that appeared alongside Melbourne House ZX81 games such as *Super Invasion*, its next-generation features stunned like a kaleidoscope vision of the future against its monochrome teletype peers.

The inclusion of the novel pushed the price to a premium £14.95 (the equivalent of the most expensive AAA game today) merited by the dazzling accompanying artwork by Greek émigré illustrator Constantine Aslanis. "We had always thought this was a good idea" Milgrom explained to Popular Computing Weekly, "because you get clues on how to solve the adventure from the book, [and] it fills many of the details we just didn't have space for." Months later, Con's fire breathing Smaug would adorn full-colour two-page advertisement spreads, with a bold headline that the game was "totally unique, with features never before seen on any computer, let alone on the Spectrum."

Sinclair Research were keen to endorse the breakthrough product, and Nigel Searle, Head of the firm's Computer Division, praised the game as one of the first programs to feature artificial intelligence. Milgrom cleverly harnessed an existing deal with Psion, a UK software developer who had an agreement to supply Sinclair with games – a relationship they'd nurtured with the successful co-publication of the *Horace* series. *The Hobbit* was promoted centre-stage [inexplicably credited to Psion with the assistance of Melbourne House] in the June 1983 edition of the official Sinclair Software and Peripherals Catalogue with unbridled praise: *The Hobbit* was a "one of the most amazing games ever written for the microcomputer."

Ports for as many British home computers as viable were soon being rushed to completion. Mitchell, assisted by Gregg Barnett [who also created *The Way of the Exploding Fist* for Melbourne House], Peter Beresford and Gerard Neïl created versions for the Oric, BBC Micro, Amstrad CPC and Commodore 64.



During an interview with Fredrik Ekman in 1987, Mitchell said "After the Spectrum I was involved to some degree with all of the other versions. I supervised all of the other versions and wrote and converted code on the Amstrad, Oric-1, PC, Apple I and Macintosh versions. In case you were interested there was also, briefly, an Amiga version that was never released. In fact, the Macintosh version was written on an Amiga 1000."

Interestingly the game was originally written for, and developed on, the Tandy TRS-80, but it was never released because the Sinclair ZX Spectrum emerged as a superior commercial target during development. Milgrom told Popular Computing Weekly that "when the Spectrum was announced it seemed an ideal vehicle, [...] the high-resolution graphics and colour make all the difference." As well as the TRS-80 and Amiga ports, another "lost" version of the hobbit was a compelling multiplayer implementation. Veronika recounted to Luke C. Jackson in one of his Storytelling interviews that "we did have a primitive version of multiplayer. At one point, we started thinking, what would happen if you had a couple of different players playing different characters, so you could have one character being Gandalf, and then overriding his behaviour list with its own behaviour list."

[Veronika] We had started thinking about multiplayer for the generic game. We'd allow multiple players to play the game at the same time, each as a different character, in essence. My thinking was that *The Hobbit* was just a specific set of databases plugged into the game engine. We could replace the databases, and you'd end up with a different game. So, any capabilities we added to the game engine would be generic.

The isolated turn-based logic of the engine was already designed in such a way that it would have needed very little rewriting to allow a human player to take control of another character. As far as Megler's code was concerned, it the way it executed instructions would be no different if it was a NPC's action list, or a human player's choices. Sadly, work creased on the multiplayer option when it looked like its continual development would stall the desired release date of the finished game.

[Veronika] We decided we needed to focus on getting the game out, rather than continuing to add cool features. It had already become a much bigger project than any of us had anticipated when we started. And we kept having ideas on how to make it cooler. At some point we just cut the scope creep.

By 1984, Melbourne House Software Inc. a US publishing office opened at 347 Reedwood Drive, in Nashville. This state-side address gave Melbourne House leverage to circumvent the copyright and licencing issues in the US, and they trusted the Addison-Wesley Publishing Company of Reading, Massachusetts to distribute the game as *The Hobbit: A Software Adventure*. They delivered on that faith in bucketloads, with retail versions for the Apple II, C64, IBM PC and Mac produced in sumptuous packaging, complete with the book, a detailed user guide, a reference card, masses of magnificent Tolkien artwork and a sublime map of Wilderland.

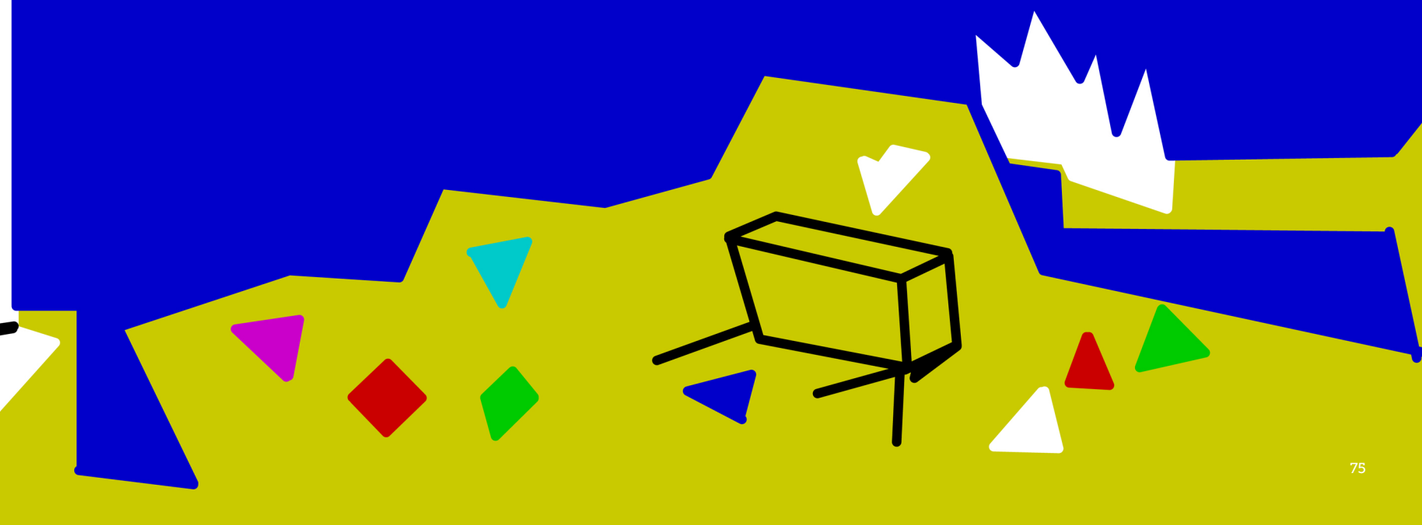
In Britain, there's little argument that *The Hobbit* was the most played, and most influential game of its time, changing the way that both player and journalist measured the standard for future games. In the final analysis, sales were reputed to be anywhere between 250,000 and a million copies making it a candidate for the best-selling adventure of all-time.

British videogame historian, Iain Mew, obsessed with sales charts, scoured the Personal Computers News [PCN] magazine archives for figures to substantiate the best-selling claim. He found that in PCN's first UK-wide, multiple outlet chart, *The Hobbit* was already a best seller, appearing at number two behind Microdeal's *Donkey King* on the Dragon 32. After that, *The Hobbit* spent two weeks as outright number one and still appeared in the PCN top ten a year later.

In May 1985, an enhanced version was released on disk, and a repackaged cassette version [including a new port to the MSX] hit the shelves in a standard plastic jewel case – this time without the book. The various disk releases featured an expanded *INGLISH* vocabulary, thirty additional locations, improved graphics [by Russel Comte and Greg Holland] atmospheric music [composed by Neil Brennen] and a tweak to the game's engine [version 1.2] to reduce the number of random behaviours and bugs. Then the US versions received a packaging redesign to correlate with *The Lord of the Rings: Game One* [published as *The Fellowship of the Ring*], *Shadow of Mordor* and *Crack of Doom* appearing across the pond.

The Hobbit's final British release came after Melbourne House and the rights to its game catalogue were sold to budget pioneer Mastertronic in February 1989. Unexpectedly they passed on re-publishing the adventure [choosing smash-hit martial arts title *The Way of the Exploding Fist* for their Ricochet label] and licenced a budget collection titled *The Tolkien Trilogy* – which bundled *The Hobbit*, *Lord of the Rings Game One*, and *The Shadows of Mordor* [no *Crack of Doom* for the UK market – **See Special Edition**] to publisher Beau Jolly in November.

Exact sales data is incredibly hard to quantify, and various sources have made academic guesses without the availability of firm sources.



Perhaps the candidates best placed to speculate on numbers are the man and woman that may have been privy to transaction or even royalty statements – if such a deal had been agreed.

[Veronika] Phil and I did ask at one time. Alfred turned us down.

Veronika herself in There and Back Again stated "The game sold in many countries over many years, and by the late 1980's had sold over a million copies". Phil Mitchell remarked to Chris Seeman in an interview for Other Hands, that "I don't have any detailed regional sales figures; though I can tell you that, across all platforms, *The Hobbit* sold in excess of a million units worldwide and the *Lord of the Rings* games did about half that combined."

For years its commercial shelf power and player appeal remained. The 1985 revamp with its reduced-price tag and 1989 compilation [at budget price] introduced the game to an appreciative wider audience. With its longevity, three distinct releases, available for almost every home micro and coupled with its US success it would be more than fair to endorse that million selling mark – and confirm that it surpassed anything achieved by the mighty Infocom. Megler and Mitchell did an incredible job compressing such an epic narrative into the confines of the 48K Spectrum. Of course, huge amounts of Middle Earth's topography and swathes of the book are compressed into a few locations, and it becomes more a series of set pieces than a world to be explored and travelled. There's no doubt the game would have benefited from an added sense of scale, time [a pronounced day/night cycle would help prepare the player for the encounter with the Trolls] and space to take a breath, even if some of the locations would be superfluous and passed without incident.

Then there's its respect of canon. In addition to placating the publishers, the book was included as the ultimate hint sheet - but it's of little help in so many areas. Despite the broad artistic brush strokes taken with the timeline, the puzzles and NPCs play fast and loose with the narrative. We're missing the rest of Bilbo's companions, having to settle for a single Dwarf in the guise of Thorin, and there is the odd implementation of Gandalf. As the manual and subsequent hint books testify, Gandalf, a member of the wise and powerful council of Wizards, has been bizarrely reduced to a doddering old fool, who muddles around without purpose, randomly grasping and examining objects.

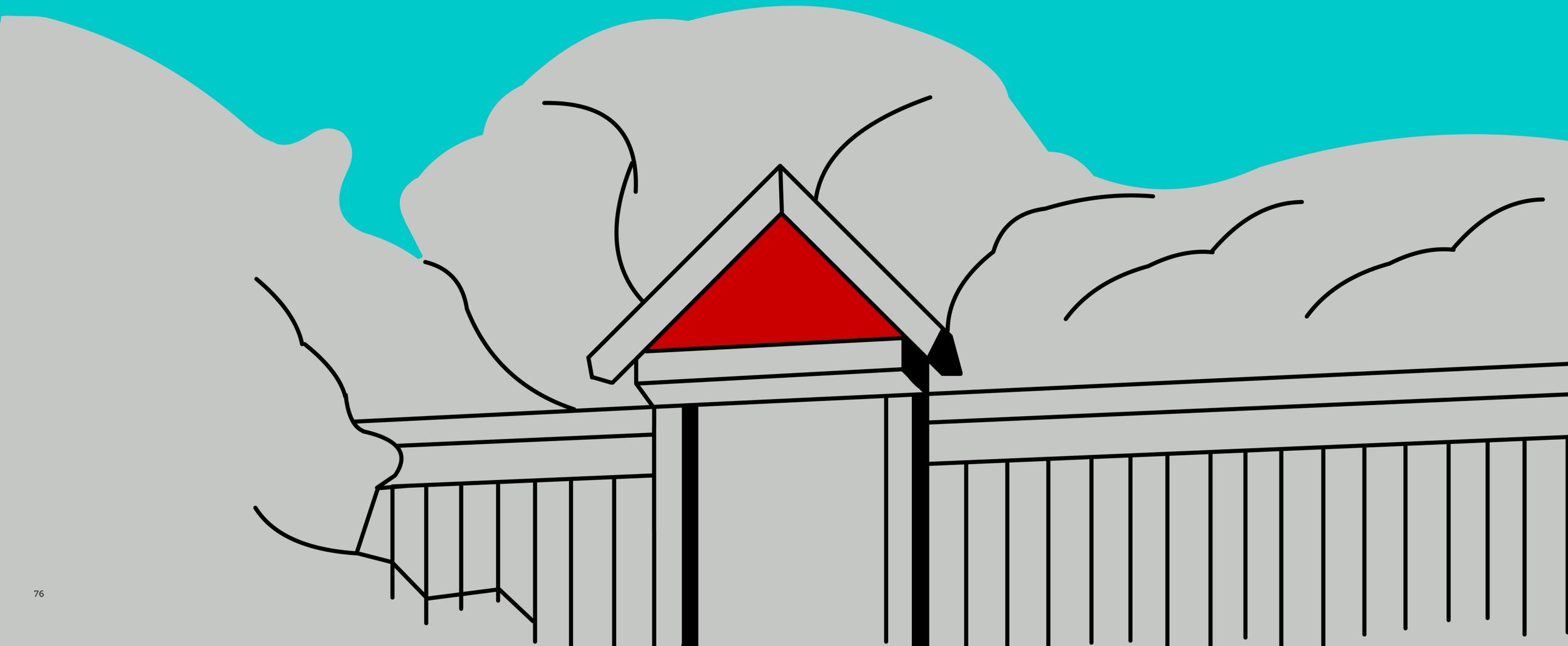
Elrond's help is essential to read the map, but much of the book's satisfying character interaction is missing. And finally, in many hints and tips texts, perhaps the furthest departure from the book, you're encouraged to kill Gollum at the earliest opportunity.

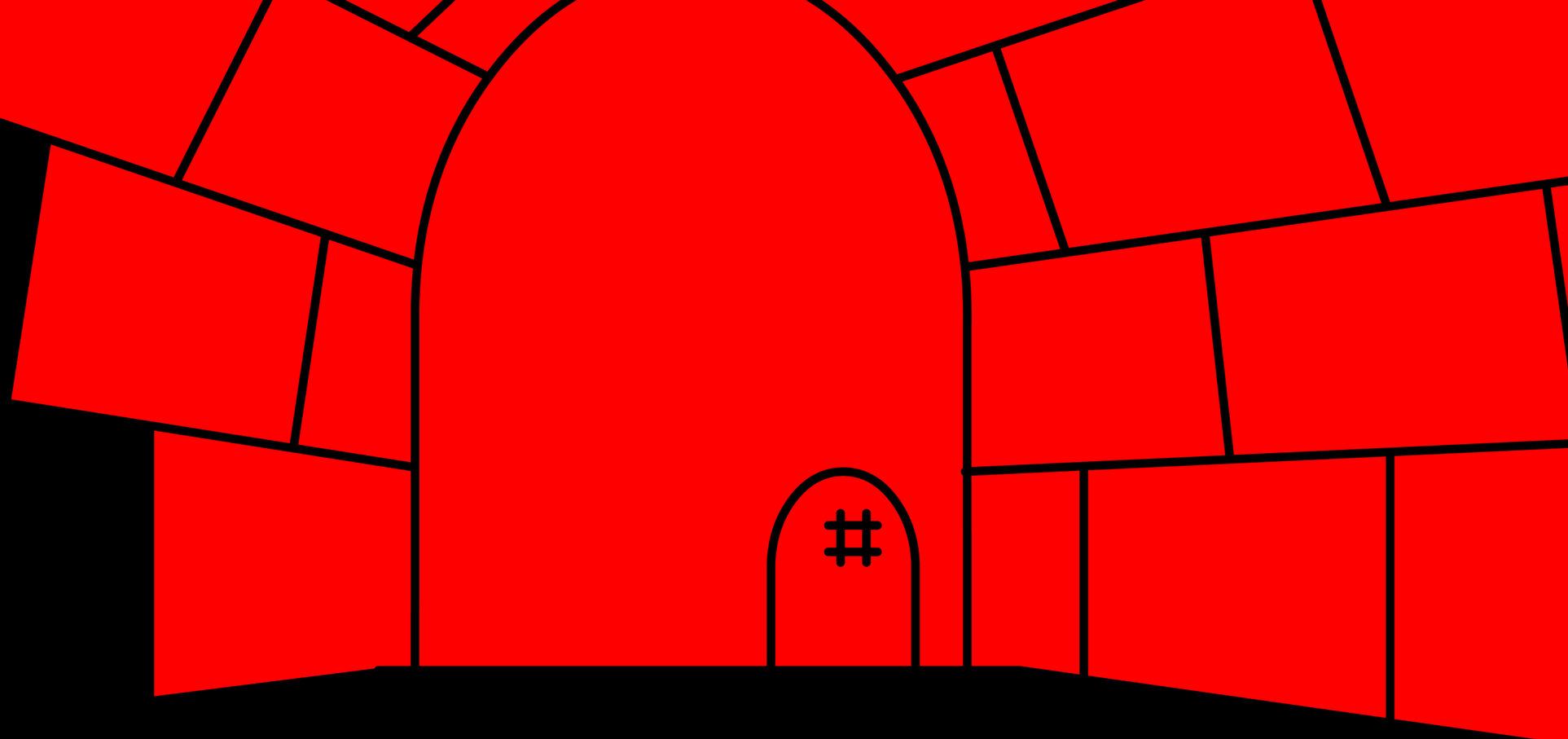
But it is here, in Megler's dynamic and lawless implementation of the rules and their ability to spit out unforeseen events that we witness the game's aforementioned soul. You're playing The Hobbit, but you're not. You're blessed with the freedom to cast aside the straitjacket of Tolkien's linearity and play a new adventure, a new story over and over again that is filled with the same familiar locations and characters.

Fans of the game recount the endless hours poured into adventures, and the unique experience that each load of the cassette would bring. Joe Pranevic of the Adventurers Guild recalls "occasions when Elrond would be captured by the Goblins, where a Warg was imprisoned with me and times when Gandalf or Thorin would be around or not." Jason Dyer, acclaimed text adventure chronicler for his Renga in Blue blog explained that it encouraged experiment, and the usual lack of replayability in normal adventures was brushed aside. "I truly wonder if there's some method through that avoids picking up the One Ring at all [and] I how far I can take the "multi-command" trick with NPCs and if can have Thorin go grab the ring for me while Bilbo just hangs out in comfort. The slight train-wreck experience, including key NPCs just randomly being dead when you need them, is actually beneficial to the feel of the game as world-toolkit, where it doesn't matter if you're trying to win but to see if Thorin can do hobbit-tossing."

It all aligns with the broader scholarly view that *The Hobbit* is a "game of emergence" where the joy comes from interacting with an unpredictable world rather than just following a one-dimensional path to the end. However, emergent behaviour or storytelling is a claim that has recently been challenged by scholars such as Jesper Juul. Since Veronika programmed the rules and actions to be random and dynamic then this behaviour should be expected, and expectation was something that went against the definition of emergence which is "neither anticipated by the game designer, nor is easily derivable from the rules of the game."

[Veronika] Saying the NPCs aren't emergent because I designed their components confuses rules with outcomes. Emergence doesn't mean 'no design at all' — it means that macroscopic behaviours arise that were not explicitly specified by the designer. I wrote low-level





rules. The higher-level game patterns and impact on the game affect and on the player of NPCs running around the game playing with each other, of appearing to players as real characters with intentionality, are the system's unpredicted outcomes. Designing the ingredients is necessary for emergence. The key question is whether the system-level phenomena are novel, irreducible, and not directly programmed. In *The Hobbit*, they are. Just like the movement of a shoal of fish or flock of birds looks cohesive but emerges from simple rules of movement relative to each other.

It's the mayhem and unpredictability that's remembered by the players, not the graphics (that are basic, and painfully slow to render with its dreadful spill-fill routine) nor the fixation with *INGLISH*. No, it's remembered for being able to get into the chest, to attack Gandalf and have your skull "cleaved", killing a goblin by hitting it with a dead troll and for Thorin's incessant need to sit down and sing about gold. Gareth Pitchford [**see elsewhere in this issue**] fondly notes "I still have vivid memories of the game, while other [adventures] from the same era have faded to the point where I may recall the names but not much else."

The preoccupation with the parser, both in development and in the post-press and promotion of the game was, in The Classic Adventurer's opinion, to the game's detriment. Milgrom's "best adventure game ever" would never be the one with the biggest number of words in the vocabulary, nor the one with the ability to string 5 command sentences together on one input line.

[Veronika] In the marketing [it was], certainly. I think that was largely due to the fact that I was gone by then, and Phil was still there, and was the person being interviewed. I believe that if you'd kept *INGLISH* but taken away the game engine and NPC innovations, that the game would not have been the hit it was. On the other hand: I designed the game to solve what I saw as the flaws in the original *Adventure* that made the game boring the instant I'd solved it, and some of my frustration in playing it and having to hit on the specific word that the author had programmed. So, I saw *INGLISH* as an integral part of the game.

There's nothing more frustrating than a clumsy parser implementation, but in reality, *The Hobbit* seldom used the sophistication it was programmed to support. A quick scan of David Elkan's official guide shows that besides the SAY TO command you rarely had to move away from simple verb-noun pairs or single verbs [around 70% of actions] to complete the game. The historical perspective on whether the time spent on Inglish was worth it is mixed, depending on whether you looked at gameplay necessity or impact on the genre.

[Veronika] Yes, but a key part of what Phil built was a flexible and generalized message generator. All messages the game generates are coded as patterns. I could simply select a message and provide a pointer to an object and a subject, and the correct message would be generated on the fly, with the correct endings. The message itself is built from words in the same dictionary used for parsing. Since only 5 bits are needed for ASCII, the other 3 bits were used to encode the information needed to make the correct adjustments. This is why you can't dump memory and use the messages to hack the game. I don't know how much of the code is common between the message generator and the input understanding, but certainly the learning was, and the vocabulary.

Whether its functionality was marketing myth or reality, the games instructions encouraged the use of adverts like VICIOUSLY or SOFTLY. Interactive fiction historians like Aaron Reed note it is "a bit hazy even today" if the game actually paid any attention to them at all. You also had to consider the amount of scarce RAM that the dictionary consumed. Reducing the vocabulary or the complexities of *INGLISH* may have freed resource better spent on expanding the location narrative, adding more locations or making the NPCs more powerful - something we did get a flavour of with the disk versions.

ATTACK CHEST WITH SWORD

Version 1.1 of *The Hobbit* shipped with a raft of glorious random behaviours and bugs that were quickly fixed in the 1985 v1.2 patched release.

Typing OP DO resulted in a set of weird results. Either evaporating the waters in the black river and fast river, or cleaving the "crack" to death. Weight had been incorrectly implemented too, enabling Bilbo to pick up a range of characters and walking around with them.

The chest appeared in many anomalies with players reporting the ability to get in it, close the lid but continue the adventure. What about tricking Thorin to get in the chest? Or attacking it with the sword: "With one swift blow you cleave his skull. The chest is dead."

Many of these unforeseen actions caused the game to crash, and a frustrating reboot of the computer would follow. "[I] discovered that the assembler had optimized away a necessary register increment, causing an infinite loop" explained Megler.

[Veronika] What would I have done? What I did! The memory usage was balanced, in that we included the vocabulary relevant to the actions and objects of the game. Making the NPCs more powerful or adding more locations would have needed more vocabulary, so the game engine and dictionary expanded in a pretty even relationship.

It wouldn't be until 1984 and Mitchell's and Melbourne House's next game, *Sherlock*, that *INGLISH* was arguably utilised to a greater and better degree. Based upon the exploits of Sherlock Holmes, it was a game discussed at Beam shortly before Veronika's departure.

[Veronika] The Sherlock Holmes series did not have the copyright challenges that The Hobbit had, and that was a major attraction for Fred. I did not believe the NPC engine I'd written in Assembler was powerful enough for the kind of interaction that *Sherlock* would need. The story of Sherlock Holmes is centred around interpersonal interactions; it seems to me a fundamentally different genre from adventures. The player interactions of The Hobbit were under-designed for complex interpersonal problem solving – there were too many special cases and special coding. I felt it required a rethink and redesign with those capabilities in mind, and that it would require a higher-level language such as C to make it feasible.

When developing *Sherlock*, Mitchell looked to tie up *The Hobbit's* loose ends in the code. "Looking back on it now" he told Computer Answers, "Thorin was a pretty boring person; he follows you around and sings about gold." He delivered NPC personality and prose but constrained their behaviour, whereas Megler had envisioned a progressive future, where the engine could self-generate characters and give them greater power.

We're left to wonder what could have been if she'd had stayed around. It's remarkable, and somewhat sad that for so long she was completely isolated from the game's success. Having been paid a flat rate of $10 an hour to create her masterpiece, there was no timely thud of a large royalty cheque every month to underline her achievement and its popularity with the public. She was abandoned from a corporate point of view, virtually airbrushed from the game's corporate credits on its release. In way, it's understandable that Melbourne House took that action. Many other games around the time had anonymous developers [who remembers David Ward crediting himself in every early Ocean release] and not having one of the game's main designers available to talk to the press and conduct interviews may have proven difficult from a PR standpoint.

[Veronika] Absolutely. From a corporate perspective, it makes complete sense. However: it's also consistent with how women are not given credit in joint projects. You see it now in the US: if a woman is in a position, it's clearly because she's a diversity hire who stole the position from the best person. Not because she's the most highly qualified, worked super hard, and earned the position. And that's what I struggle with: if the thing that "makes perfect sense" excludes the people who deserve credit, when, and how, do you account for and correct that?

```
You open the wooden chest.

You put the valuable treasure in the
wooden chest.

a cheering crowd of dwarves, hobbits and
elves appears. Led by gandalf they carry
you off into the sunset, proclaiming you
hero of heroes and master adventurer !!!.
```

Melbourne House were to repeat the same error with *The Hobbit's* follow-ups. As they expanded the licence into the full *Tolkien Software Adventure Series*, the individual identities of the talented designers and coders responsible were buried behind a credit to The Faceless Programmers Cooperative. In an interview with The Register, Megler reflected on what the impact on her own life would have been if she'd been aware of the game's success and been fully credited with its triumphs.

[Veronika] The one time I'd mentioned the game at a party in the mid-80s, someone told me that it was a bad game. After that I never mentioned it again. Knowing otherwise would have given me more confidence that my ideas were worthwhile, more resilience when I was going through tough periods in my career.

Thankfully, with the growth of the internet and the vibrant retro community, Veronika started to receive emails from devotees across the world and conducted her first interview for an Italian fan site in 2001. In 2009 while studying for her PhD, she was recognised by a fellow student who played the game as a child in Mexico. That enthusiasm led her to present her work to the faculty, sparking her own further research into the game's global impact. The retro community interest escalated, and as *The Hobbit* celebrated its 30th anniversary several retrospectives were published, including a comprehensive delve into the designers by the Play It Again perseveration project and a controversial article by The Digital Antiquarian - Jimmy Maher.

[Veronika] I tripped over a blog post talking about what a crazily brilliant programmer Phil had been and credited him with all my work as well as his own. I got angry and wrote to the author. He ended up challenging me, then became convinced and revised the blog post giving me credit for my contributions, to his credit. But even then, the revised blog post writes about how weird it was that these ideas came from a woman. There's no suggestion of me being crazily brilliant. Various comments challenged the blog rewrite, and the case was closed when someone who'd spent years reverse-engineering the game from the Assembler code commented that the game was clearly split between game and language engines, with a clear interface, and the coding styles were very different between the two, implying different authors. That's when I decided I needed to step up and start correcting the record.

Publishers such as Retro Gamer, Bitmap Books and Read-Only Memory, combined with reputable academic study and journalism from people like Helen Stuckey, Aaron Reed and the Australian Centre for the Moving Image are doing sterling work in correcting the historical record. It's important that the decades-long narrative that erroneously credited Mitchell as the game's primary creator while marginalising the role of Veronika Megler is challenged.

[Veronika] There were a couple of interviews with Alfred – one as recently as 8 or 9 years ago – where he described me as having only converted the book to the game and written some of the puzzles. When he should know better; after all, he was there; and now, so many years later, there's nothing to gain. As far as I know Phil never claimed credit for more than INGLISH, but other people have ascribed the rest to him.

Maher was happy to put the record straight in his *The Hobbit* Redux post citing that he understood that Milgrom and Melbourne House press and marketing materials vastly overstated Mitchell's role primarily to build "genius" branding around Mitchell to promote the forthcoming *Sherlock*.

In summing up, what cannot be recaptured in playing the game, or fully appreciated by the thousands of words written about it, is what a monumental paradigm shift and epoch defining moment the arrival of *The Hobbit* was, and how early in the history of the home videogame industry it came. Arcade games like *Pac-Man*, *Space Invaders* and home micro gems like *Manic Miner* [arriving in 1983] and *Elite* [in 1984] set a benchmark and where genre defining. *Miner* and *Elite* are the British videogame equivalent of the Mona Lisa - true masterpieces - and *The Hobbit* should be added to that gallery alongside both. It became the gold standard of adventure design that arguably was never matched [especially in cassette-based Britain] until the arrival of the 16-bit era, and the more complex games and systems from Magnetic Scrolls and Level 9.

Its rightly cited in many greatest games of all time lists, and has been the star in museum exhibits, a preservation project and a whole raft of podcasts and publications, including Retro Gamer Magazine, and the 1001 Videogames That You Must Play Before You Die book. Megler herself was the star of Great Big Story's "The Hunt for the Hobbit's Missing Hero", part of their Emmy® nominated series of videogame documentaries. The short film is a fitting tribute to Veronika's extraordinary legacy.

[Veronika] Thank you! It was very special to me to have the producers reach out to me, and to come and film. It's been an honour to discover how many people's lives I've touched. It's been a privilege to have people track me down to tell me the impact the game has had on their lives. People became interested in interacting with people rather than shooting them in shooter games, or puzzle solving, or computers, or linguistics. Many people worldwide have told me it caused them to learn English or sparked their interest in reading. One person told me he went from reading Enid Blyton level, to reading Tolkien. I recently had a colleague from Spain tell me he shocked his English teacher in school at the time by knowing the word "portcullis"!

Megler has contributed to some amazing technology throughout her career, including *Amazon Web Services* that power the modern internet and which in raw numbers may have impacted more people's lives in ways that they're not really aware of.

But, with *The Hobbit*, it's where she'll be thankfully and fondly [and probably immortally] remembered for her contribution.

[Veronika] I feel touched, and chastened. The randomness of it: I happen to answer a tiny ad on a billboard at school, and I happen to be the one who walks in and is given that instruction, "write the best adventure game ever". It's also pretty strange to be at the other end of a long, varied and colourful career, and have the most widely impactful thing you've ever done be something you did as a part-time job in university. My fifteen minutes of fame, and I wasn't even aware of them. It's like the PhD I did in my 50s: my most-referenced paper, with nearly double the citations of any from my research, was a side-effect of a project I did for a GIS class (Spatial analysis of graffiti in San Francsico). But there's plenty of time for me to do some other random, impactful thing! I'm super curious to discover what it'll be. There's always time for another adventure!

GARETH PITCHFORD

Satirical adventure designer and Twilight Inventory author **Gareth Pitchford** wrestled with the intricacies of *The Quill* and *GAC,* before finding a spiritual home for his sense of humour and writing talent with Scott Denyer of Delbert The Hamster Software.

The Hobbit **was probably the earliest adventure game I can remember playing. Both my Dad and I were big fans of Tolkien, so it was one of the few full price games that we bought new. The inclusion of a copy of the novel undoubtedly sealed the deal!**

To help us complete the game, we picked up a copy of David Elkan's 'A Guide to Playing *The Hobbit*'; which must have been one of the first videogame strategy guides created. It seems there were a lot of people who were very keen to find out how to get out of the Goblin's Dungeon.

When were you given a Spectrum – it seems to be your love?

We had started off with a ZX81, [...] not long after, we picked up a 48K Spectrum from Lewis's, when it was on offer with the 'Six Pack' [of games]. [We] eventually upgraded to a Spectrum +3. My Dad and I clubbed together to buy it through the Grattan catalogue... paying back the money over 100 weeks... with plenty of interest added on, of course. It must have cost a fortune. It was worth it, though. Having a disk-based system was very convenient for writing and playing adventures. By 1992, I'd picked up a Sam Coupé and that became my new 'Spectrum' of choice.

What other mainstream adventures did you play?

Aside from early adventures like *The Hobbit*, *The Lord of the Rings* and *Valhalla*, I didn't really start appreciating and getting into adventure games properly until the latter half of the 1980s. I eventually realised that adventures were something I really liked, and could actually complete, through playing the examples included on the covertapes of magazines such as Sinclair User, Your Sinclair and Crash.

Incentive's *Ket Trilogy*, the *Legend of Apache Gold*, *Winter Wonderland* and *Karyssia – Queen of Diamonds* were all showcased on the Crash covertapes. And on the Your Sinclair side of things, *Red Door*, *The Gordello*

Incident and *A Harvesting Moon* gave me my first sample of more indie, homebrewed adventures.

Did these pique an interest in writing games?

I'd always found the adventure columns in the mainstream magazines very interesting, particularly Mike Gerrard's writing in Your Sinclair, but now I started actively playing the games, writing in to the columns, ordering from the indie publishers and offering help to other gamers who were stuck in the adventures that I'd completed.

On the subject of Mike Gerrard's column, I think one of John Wilson, of Zenobi's, masterstrokes was putting together the first *Best of the Indies* compilation for Your Sinclair readers. That first discounted cassette, featuring Tartan's *Double Agent*, Linda Wright's *Cloud 99*, Terry Taylor's *The Labours of Hercules*, Jack Lockerby's *Domes of Sha* and two of Zenobi's own title was surely responsible for introducing a lot of players to the range and quality of homebrewed adventure games.

What impact did *The Quill* and its peers have on the industry and for indie creators such as yourself?

The Quill, *GAC* [*Graphics Adventure Creator*] and *PAWS* [*Professional Adventure Writing System*] removed the barrier of entry for wannabe adventure game creators. They allowed adventure players to easily become adventure authors, producing fast, feature-rich and responsive games without needing detailed knowledge of assembly language. And once the game was created, authors could freely distribute their adventures without having to worry about permissions or license fees.

The decision by Gilsoft particularly to omit the need to pay licensing or royalties had big repercussions didn't it - how different could the landscape have been with a more costly barrier to entry?

Without the various game creation tools that allowed hobbyists to easily produce their own titles, I think text adventure games would have died out as soon as the big publishers decided they weren't making enough money from the genre.

You originally used *GAC* but then switched to *The Quill* and eventually *PAWS*. What differences were there between the platforms and why did you switch as a programmer and writer?

Even though *GAC* technically supported more locations, *PAWS* allowed you to create bigger adventures, even for 48K machines, because of the way the creation program used overlays to load in parts of the adventure writer as they were needed. *PAWS* had a pretty effective built-in text compression utility [...] and an option to create adventures specifically for 128K machines; *GAC*, on the Spectrum, was 48K only.

The fact that you could create your own subroutines, using the *PAW's* user-defined process tables, made that package more powerful and elegant as a programming tool. There was the ability to call external routines (which savvy programmers could use to load graphics or produce visual & audio effects). Useful quality of life features, like RAMSAVE and RAMLOAD were available as standard. *PAWS* also made altering the look and feel of adventures very easy; allowing authors to lock some text in place and allow other parts to scroll underneath. This meant you could have the location description on screen at all times, should you desire, so players could easily refer to it and not have to constantly REDESCRIBE the location. It wasn't so much that *PAWS* adventures all looked very different, in reality they often used similar screen designs, it was the fact that they looked different to "old" adventure games. *GAC* games, with their vertical scrolling and the often-used default font, tended to look and feel very dated indeed.

Your first adventure, *Microfair Madness* was released in 1991. We are beyond the Spectrum's commercial life, and most gamers had moved onto the ST or Amiga, why didn't you?

I never had a 16-bit machine so writing an adventure for 16-bits was never an option. Saying that, even adventure playing friends, that moved to the next generation, never seemed satisfied with creation utilities like *STAC*, preferring to continue to write and play titles on their 8-bit computers.

It was the late eighties when I began seriously exploring the idea of writing my own adventure games. In the past I had borrowed some adventure creators, like the *GAC* and *The Quill*, off friends. These copies... well, how shall I put it... came without all the necessary documentation... so they just made me more even more baffled as to how to start.

I invested in a copy of Tartan's *Adventure Builder System* to see if I could produce anything with that. This was based on the code that Tom Frost used for his own adventures. In his hands, it did amazing things. In the hands of a complete novice, like myself, it was just too overwhelming, so I decided to see if I could convince someone else to program my ideas for me. I wrote in to Mike Gerrard at Your Sinclair, saying that I'd designed an adventure and needed a programmer, and he put my request & address in his column.

That led to a working relationship with Scott Denyer of Delbert The Hamster Software?

Scott had previously written *Arnold the Adventurer* for Zenobi Software and had decided to that he would self-publish his next few games. He spotted my request for a programmer, in a two-year old copy of Your Sinclair, and got in touch to ask if I was still looking for someone to collaborate with.

You designed *Microfair Madness* and Scott programmed it. Tell me more about the inspiration for the game.

Despite having never actually been to a computer fair, I liked the idea of using a convention or exhibition as an interesting setting for some strange events and a way of bringing together lots of different characters. [Its] a cross between a time-capsule and a teenage diary for me. It incorporates almost everything I was interested in or that featured in my life at the time. Whether it's TV shows like Doctor Who and Star Trek, 1950s radio comedies like The Goon Show (not something your average teenager was into back then) or the music of Jean Michel Jarre. There are puzzles based on objects that I had in my bedroom and cameos from all the various people from the adventure game scene, that I'd read about in magazines.

You included a rather nice idea within the adventure - the ability to play other games within a game, beating the *Maniac Mansion* Easter Egg in *The Day of the Tentacle* by a few years?

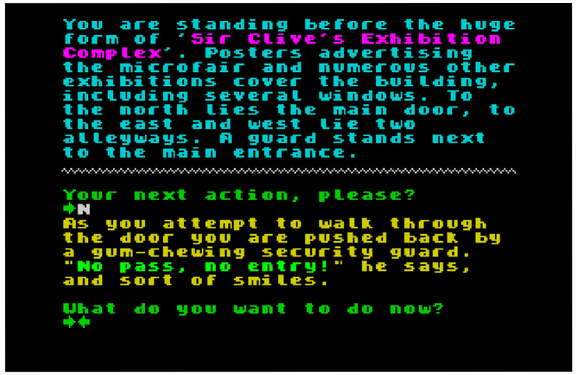
As the setting was a computer microfair, I thought it would be quite cool if you could actually transport yourself into the various adventures that the vendors were selling. It allowed me to play about with some ideas for more compact games. As the name might suggest *The Hoppit* spoofed the Goblin's Dungeon scenario in *The Hobbit*. *The Search for Smok* was a sci-fi parody, bridging the gap between the two main parts of the adventure. One of the other mini-games, *The Quest for the Holy Snail* was later expanded into a standalone release.

It was definitely a lot easier coming up with the idea of including the minigames knowing that someone else was going to have to figure out how to program them.

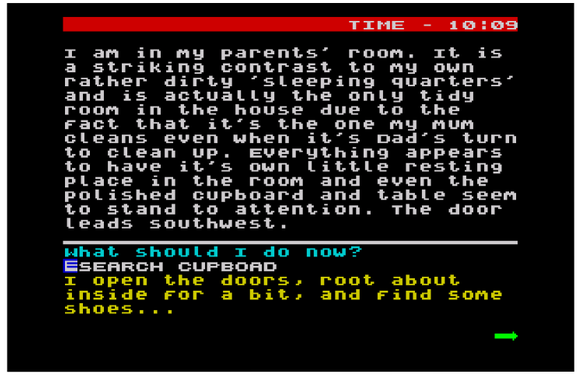
What was your own process for writing an adventure?

I used to start with pages upon pages of scribbled notes; listing ideas for puzzles, the key events in the story, potential characters, bad puns and other jokes. Usually bad jokes. Then I started drawing up maps of the locations, filling the page with annotations showing where all the objects and puzzles were to be placed, and how they linked together.

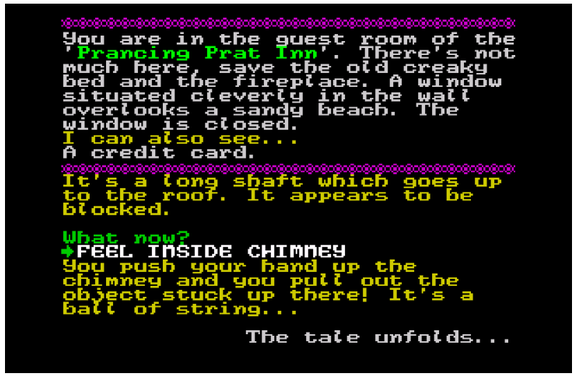
Once the adventure was planned out completely and I was happy with how everything worked I would then start on the actual text. I'd write everything out on paper first, beginning with the location text. Key words in each location were highlighted so I could refer to them to write all the additional descriptive text. Finally, I'd list and detail all the puzzles with the appropriate matching responses.



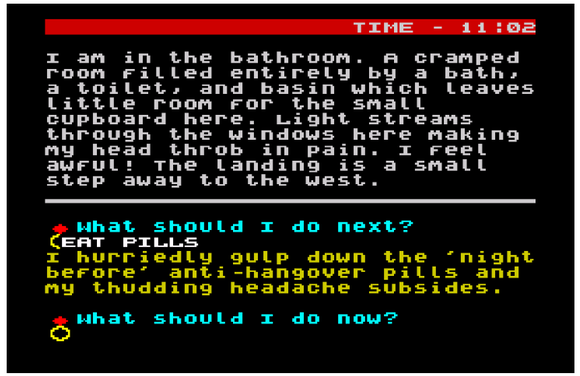
[Microfair Madness] Gareth's first, and one of his best adventures, set in and around one of the computer shows of the late 80s and early 90s. There's plenty of taking the michael, groansome puns, and oodles of references to personalities at the time, including an appearance from adventure guru Mike Gerrard.



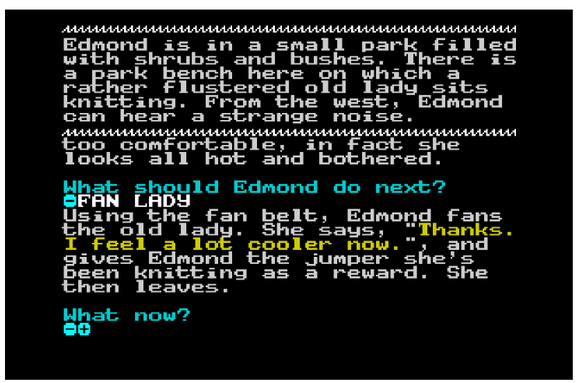
[First Past The Post] In *First Past the Post* you are Ernie Spludge, a character never far away from trouble. After falling out with his fiancée and sending her a rude letter, he quickly realises the error of his ways. It's a race against time in this tricky and humourous game to intercept the letter before its delivered and the romance is ruined.



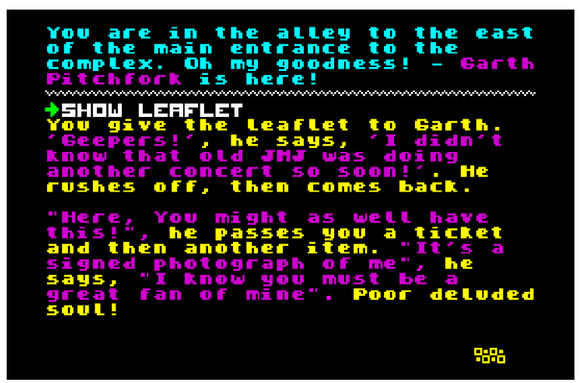
[The Quest for the Holy Snail] What is it about the Arthurian legend and adventure games? *Snail* in fact is an expanded version of a mini-game that first appeared in *Microfair Madness*. There's plenty of fun, Tolkien/Arthurian references and good use of *PAWS* too.



[Get Me to the Church On Time!] Ernie Spludge is back, and this time he's drunk - well, hungover, after a heavy stag night. Suffers a few annoyances, like having to hit a key after actions, and a broken GET ALL command - but a welcome return for the anti-hero Ernie.



[Life of a Lone Electron] Rewriting Atomic Theory, *Electron* has you playing Edmond the Electron, crash landed on the planet Elektronz and trying to escape. Nice presentation and surreal puzzles (a fanbelt used as a fan) await you in this "educational" game.



[Personal Computing Whirled!] A sequel of sorts to *Microfair Madness* that was designed as a promotional release. It was initially distributed through From Beyond PD, as well as given away as a free adventure with the Sam Coupé Adventure Club magazine.

This way of working was originally born out of necessity... I'd design the game then copy out everything by hand so it could be sent off to Scott to program the actual adventure on the computer. [...] Two of the most important qualities that Scott had were that he owned a copy of Gilsoft's *PAWS* and that he had the ability to read my dreadful handwriting. On a more serious note, Scott had a similar sense of humour, knew his way around the [utility] and seemed to have endless enthusiasm for bringing projects to life. For our games together, I did all the design and writing and Scott did all the actual programming. And sneaked in a few of his own jokes.

One of the most important parts of the game creation process was the playtesting. Each game was generally sent off to at least two different playtesters and their detailed reports of bugs, spelling mistakes & suggestions for improvements were carefully reviewed and acted upon (if possible) before the games were published.

Microfair was well received by Crash who said it was "a great game", and Your Sinclair who said "it's inventive [and] funny when it wants to be", that must have been fulfilling?

Having our adventure reviewed in the same, commercial magazines that I'd been reading for years was great. I was a little sad that we managed to miss Mike Gerrard's tenure, particularly given his role in the game's creation (and in the actual game itself). He departed Your Sinclair just as we were finishing the adventure so we managed to squeeze in an in-game reference to the move. Tim Kemp's full-page review more than made up for any disappointment. I think I managed to impress my Computer Studies teacher, who was a Your Sinclair reader, with that.

A reflection of your all-round creativity is the fact that you drew the distinctive artwork for the advertisements and produced the loading screen too?

When you were self-publishing a game you had to do all the work yourself. [...] Scott did all the inlay designs and was the poor sod who ended up duplicating cassettes and stuffing jiffy bags.

You followed the success of the game with an obligatory sequel called *Personal Computer Whirled!* - but this time without Scott?

After our first few releases together I acquired my own copy of the *PAWS* which meant we could both concentrate on our own games. *Personal Computer Whirled!* was a mini-sequel to *Microfair Madness*, released as a Public Domain/promotional title. [...] I wrote it primarily as a way of learning how to use PAWS. [...] Scott decided he really needed to concentrate on his studies at college so I published my next games through Zenobi Software.

Did you change the way you worked now you were coding?

I worked in much the same way. [...] I'd generally still draft out the location descriptions on paper but a lot of the other text was written directly into the computer as I built up the code. I now also kept pages of notes about what each 'flag' did and made sure I saved copious amounts of dot-matrixed sprocket-fed printouts of the database. Just in case the worst happened and the adventure saved to tape or disk failed to load.

Did you prefer to go down the humour approach to writing adventures? Did that suit your personality? It was one area of adventuring that was hugely popular – with St Brides and Delta 4 taking comedy and satire and parody into the mainstream?

I think I probably found it easier to write humorous games that weren't restricted by the constraints of reality. I'd never played any of the Delta 4 or St. Bride's titles, but I was heavily influenced by the work of people like Spike Milligan and Douglas Adams. I did have several more serious titles in development, which I always had a feeling would've sold better, but for various reasons they were never completed.

Do you have any rules, or "ten commandments" for others who want to write adventures?

I would just say: write an adventure game that you'd like to play. I think at times I put things in my adventures just because I thought that's what other people would want to see. Things like sudden deaths or the need to repeatedly examine or search containers. When I go back and replay my old adventures now, they're the things that really annoy me.

How long it typically take to write a game?

Looking back, it's hard to remember how long it took to create the games. When you're at school or university, playing and writing games in the evening, free time seems almost endless. I guess it probably didn't take more than a few weeks to produce a regular-sized game. *Microfair Madness,* because of the size, the number of versions and the slowness of collaborating by post, probably took about five months of work. Scott was definitely doing the lion's share of that!

You split several games into 48K and 128K variant, was this also very time consuming?

It was certainly a lot less time-consuming for me to write & design than it was for Scott to program. The original version of Microfair Madness was a two (and a half) part 48K adventure. To expand the game, I just had to come up with extra locations and puzzles. However, because of the way *PAWS* utilised pages of memory, when Scott programmed the 128K version he first had to type in everything he'd already coded for the 48K game. Producing *Microfair Madness* was basically the same amount of work as creating five individual adventures.

Ernie Spludge was a game where you experimented with your method of creation, taking inspiration from Larry Horsfield and John Wilson by sitting down without a plan and entering the adventure directly into PAWS?

[...] I decided to do a prequel adventure [...] and use it as an opportunity to try working in a slightly different way. I still scrawled down puzzle ideas and mapped things out on paper first, but the majority of the work on the game was done directly on the computer. Sadly, the disks with the incomplete code have long been lost and it would be difficult to recreate the work I'd done on it.

Looking back, which game are you most fond of?

It's hard to look back on any of my games without seeing the mistakes, obtuse puzzles and glaring omissions. When you end up going online to look up solutions to your own puzzles then you realise you could've probably dropped in a few more hints and made things a little less obscure! *Microfair Madness* is probably the game I feel I have the greatest connection to, but I think the two-part *The Search for the Nether Regions* is the title I'm happiest with.

You're a member of a popular 8-bit Adventure group on Facebook, and within it you've made several interesting posts on unreleased games you'd designed?

Like most people that dabbled with adventure writing, I have ring-binders full of puzzle ideas and unfinished adventures. Some games exist only as titles and scribbled collections of puzzles. Others are more fleshed out.

One of my first game designs was for a Sherlock Holmes adventure, set within the confines of Baker Street, where the detective had to lay a trap for Moriarty. It was full of flavour text and copious references taken from Conan-Doyle's novels, but I'm not sure it would've ever really worked as an adventure, even as a B-Side. I was a big fan of Tony Collins' adventure *Methyhel* and had started talking to him about producing a sequel to that game to form the second instalment in the *Sinister Investigations* series. I think we had quite a good premise, and some interesting initial ideas for the game but things went quiet on that front when Tony withdrew from the adventure scene.

Deception of the Mind's Eye was one of the designs I came up with after I'd advertised for a programmer in Your Sinclair. It started life as

a sci-fi murder mystery based on the beginning of a short story that I'd written. Again, I probably hadn't thought through whether the ideas I had would make a compelling adventure. I'd started work with a programmer on bringing the game to life, but we hit some snags early on and both lost enthusiasm for the project.

I liked the title, though, so *Deception of the Mind's Eye* eventually became a two-part fantasy adventure for 128K Spectrums. Part one was finished and the second part was started just before I went off to university. Unfortunately, when I came back at the end of the year my Dad had sold our Spectrum +3, leaving me with no way of finishing the tale! I kept the disks, though, and about twenty years later I managed to recover what work had been completed and uploaded it to the Internet.

Destination: Planet of Origin is probably my completest most-uncompleted game. Looking through my plans, I seem to have finished all the design work for two out of the three planned instalments of the adventure. It was another serious sci-fi game, where you took on the roles of three members of an exploration team searching for the planet of origin of the human species. Each character in the game had their own areas of expertise and abilities, so you had to switch between them and work together to solve the challenges you were presented with.

I was partnering with a fairly inexperience programmer on that title who found implementing the multiple character system quite difficult, so things never really progressed further than my notes.

Moving onto your recent book, Twilight Inventory. Why did you think the time was right to publish a title on niche Spectrum adventures?

I'd been planning to write a book on text adventures for several years but had never quite decided on the scope or the exact focus. I really wanted to write something about the British adventure scene. There had already been a lot of articles and books on early commercial adventure games but hardly anyone had written about the adventures that I remembered playing; the independently produced, home-grown adventures of the late eighties and early nineties.

Those are the stories and experiences that are in danger of being lost and forgotten. Memories fade, disks and tapes fail. Several of the authors who produced the games are no longer with us.

[...] I decided to put together [...] something I could complete quite quickly, using my archive of existing reviews that I wrote for various adventure fanzines in the 1990s. Something that would get the conversation about old home-grown adventures started.

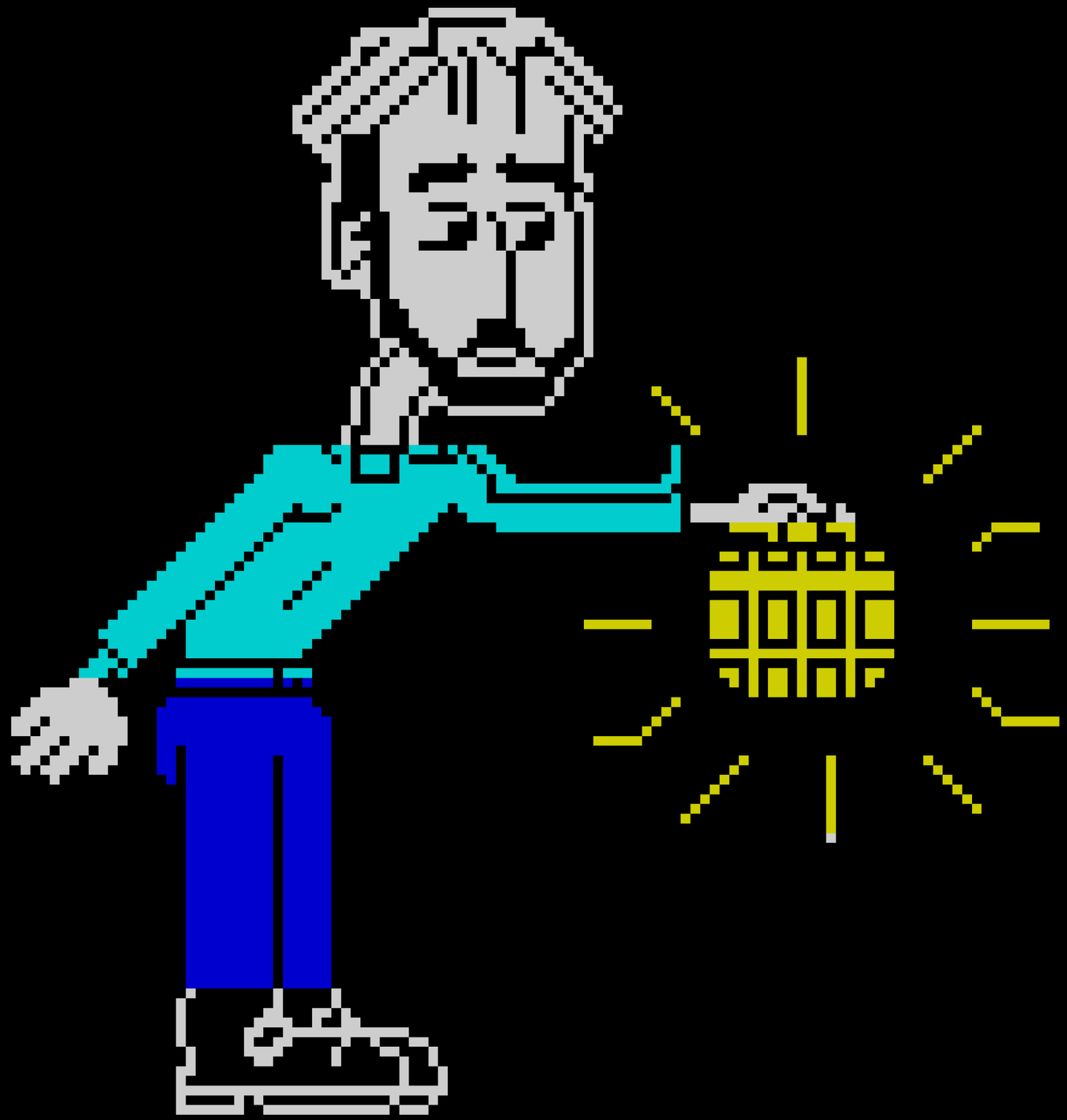
Did you approach any publishers with the idea?

Oh no. I could only ever see Twilight Inventory appealing to a very, very small audience. A book on text adventures for the Spectrum would only interest a small number of people. A book just on homegrown text adventures, mostly produced after 1990, is just about as niche a niche as you can possibly get. Another issue is that text adventures aren't hugely visual. It would be a lot easier to convince a publisher that a book with glorious, 1980s pixel art is a good idea than it is to sell a book with pages upon pages of screenshots of blocky writing.

I loved the cassette style artwork for the book, though probably not noticeable unless you view the PDF?

I wanted the book to look like a homegrown adventure cassette box, so I deliberately used a very rough doodle (as if sketched out on lined paper) as part of the cover.

The back of the book is styled like the reverse of a tape box, with a sticky note attached, and I even popped one of those gold 80s-style address stickers on the front too. I was tempted to use some



Letraset transfers or embossed Dymo labels. Perhaps I'll dig those out if I ever do a second volume.

How successful has Twilight Inventory been, and what has been the response from the ever-growing adventure community?

When it was released there was a comment from someone on one of the Facebook groups about how they could only ever see it getting single-digit sales! It exceeded that target within the first day, thankfully! I am very pleased with how it's done so far. It's been great seeing people post pictures of the book on social media, especially alongside their retro gaming collections. It's nice that it seems to have got people talking about some of the later indie titles, as that was the whole point of writing it.

You've omitted your own games, why?

Yes, I deliberately didn't mention them. There were far too many other interesting games to write about and it didn't feel right to review my own titles. I'll leave that job to other people.

You've also made a brave decision to release a digital download for free, which I think is commendable.

[...] I wanted to give everyone a chance to see if it was worth spending their time or money on. The option is there to pick up a printed version or Kindle edition for those that want it. I've been surprised by how many people actually have.

Can you speak a little more on the legacy of such publishers as Zenobi, FSF Adventures and The Adventure Workshop who really supported this community beyond the commercial life of the Spectrum?

There were lots of adventure authors who published their own games but companies such as Zenobi Software, The Guild, and The Adventure Workshop, gave those people, who didn't want the hassle of dealing with postal orders and stuffing jiffy bags, a way to release their wares to the wider adventuring community. As an adventure author, it was nice to have all the business side of things taken care of and, as an adventure player, it was also convenient to have a reliable, regular, trustworthy source of new experiences.

I think it's important to remember that, although it started out as a homegrown enterprise just selling his own games, John Wilson's Zenobi Software was, for all intents and purposes, a commercial company. John did everything in a professional way with proper contracts, a generous signing-on fee and regular royalty payments. Zenobi was run as a business, with all the bookkeeping and tax obligations that it required. He also set the standard for customer service and proved that, by delivering quality titles to a dedicated audience, text adventures had a far longer life than the mainstream publishing companies had predicted.

As you mentioned, the book is an updated compilation of reviews you undertook in the 1990s for various fanzines - these were key in driving sales of games and maintaining a community.

Although each commercial magazine and 8-bit machine had its own community, and often its own fanzines, Adventure Probe was the central hub of the British 8-bit adventure scene. Throughout its twenty-one years of existence, it acted as the forum and focus point for the hobby. Its extremely busy letters pages were filled with news, discussions and lively debate, functioning much like social media and forums do today. It was a real community magazine where people formed friendships. People would join as readers and often go on to be reviewers, playtesters and adventure writers themselves.

What about the conventions?

The Adventurer's Convention was an annual UK event set-up by several readers of Adventure Probe magazine. The first event, in 1990, saw over a hundred people gather together in a hotel in Birmingham where there were indie publishers selling games, 8-bit and 16-bit adventures to play, an awards ceremony and the all-important chance to finally put faces to names. The first event was such a success that the convention continued to be held regularly each year with the second event, in 1991, attracting over 200 people. The highlight of each Adventurer's Convention was arguably the Megapoints competition where attendees were given a set time to play a specially written adventure to see who could score the most.

The book highlights the importance of women both as authors and players in the community. What do you think of the legacy and impact of mainstream names such as Anita Sinclair and Veronika Megler and female indie authors such as Kez Gray and Linda Wright?

It would certainly be true to say that, in those adventures where you were cast in the role of a named character, that character was often male. However, the player was more usually assigned the role of a genderless "you". The lack of an on-screen avatar, as was common in arcade & action titles, meant that it was easier for adventure players to project themselves into the game.

I don't know if that encouraged a more diverse audience, but adventure games seemed to appeal to people of all ages and from various walks of life. There was certainly a high proportion of women involved in the community. A lot of the fanzine editors, reviewers and playtesters were female. Some of the best examples of games from the 8-bit indie scene were written by women, such as Linda Wright, June Rowe and Sue Medley.

When you went back to edit the texts, did you change anything, or change your mind about any reviews?

Most of the edits I made were to add a little variation to reviews that were originally designed as standalone pieces. I was very keen that the reviews should remain a contemporary assessment of the titles although I did go back to replay some of the games to add extra detail to a couple of the reviews, particularly those where I felt my original judgements were a little too harsh.

Finally, do you plan on writing any new games, or finishing off some of those games that you left on the drawing board. There's the prospect of a new *PAWS* for the Spectrum Next computer?

It's hard to get too excited about the idea of spending a lot of time writing a brand new adventure game that hardly anyone would play. I think that sometimes people are more enthusiastic about talking about the old games that they remember playing, than they are about seeking out and spending time trying new experiences.

I doubt I'd be able to afford a Spectrum Next any time soon, though. It does look like a neat machine, though. It reminds me a lot of my old Sam Coupé. [...] Even today, [*PAWS* is] still a brilliantly simple but powerful programming tool. [...] I'd put on my wish-list [...] an improved editing & input system, and better memory management.

8-BIT BRITS

Gareth is a formidable text adventure researcher and continues to do sterling preservation work tracing authors and documenting early examples of British games.

He is an editor on the Classic Adventure Solution Archive and regulary contributes to a variety of adventure related forums including Stardot and IFWiki.

His 8-bit Adventure Games website has an abundance of information, including interviews and profiles, adventure writing resources and historical notes.

https://8bitag.com

DESERT ISLAND DUNGEONS

Gareth Pitchford is thrown overboard as his ship runs aground and begins taking on water. Wading ashore of a handily placed desert island, his helpful friendly hamster, Delbert deserts the ship with five adventures strapped to his back.

The Final Demand by Steve Clay on ZX Spectrum
[Part of] Steve Clay's excellent *Taxman* trilogy, where he cleverly combined traditional adventure game fare with Crystal Maze-style brainteasers and puzzles.

Agatha's Folly by Linda Wright on ZX Spectrum
Although *Cloud 99* is probably my favourite Linda Wright game, for my desert island trip I'd choose this longer adventure. *Agatha's Folly* features two contrasting parts, both expertly written and well-worth playing through again.

Dr Jekyll and Mr Hyde by The Essential Myth on ZX Spectrum
Something to keep me busy while I waited for rescue. This gothic three-parter, from The Essential Myth, is perfect. It was based on the book by Robert Louis Stevenson and I got stuck in the first section for months on end.

Brian & the Dishonest Politician by Scott Denyer on ZX Spectrum
Trying to win votes in an election is a perfect excuse for the usual series of adventure game quests, [and] Scott had the cheek to cast me (or someone with a very similar name as me) as the evil, slimy antagonist!

The Thirty-Nine Steps by Jack Lockerby on ZX Spectrum
There's no point just taking games to the desert island that I've already played, so my last game is one of River Software supremo Jack Lockerby's final adventures. Jack had written very enjoyable outings based on classics like Treasure Island, A Christmas Carol and Kidnapped in the past so it would be interesting to make time to play his take on the John Buchan adventure novel.

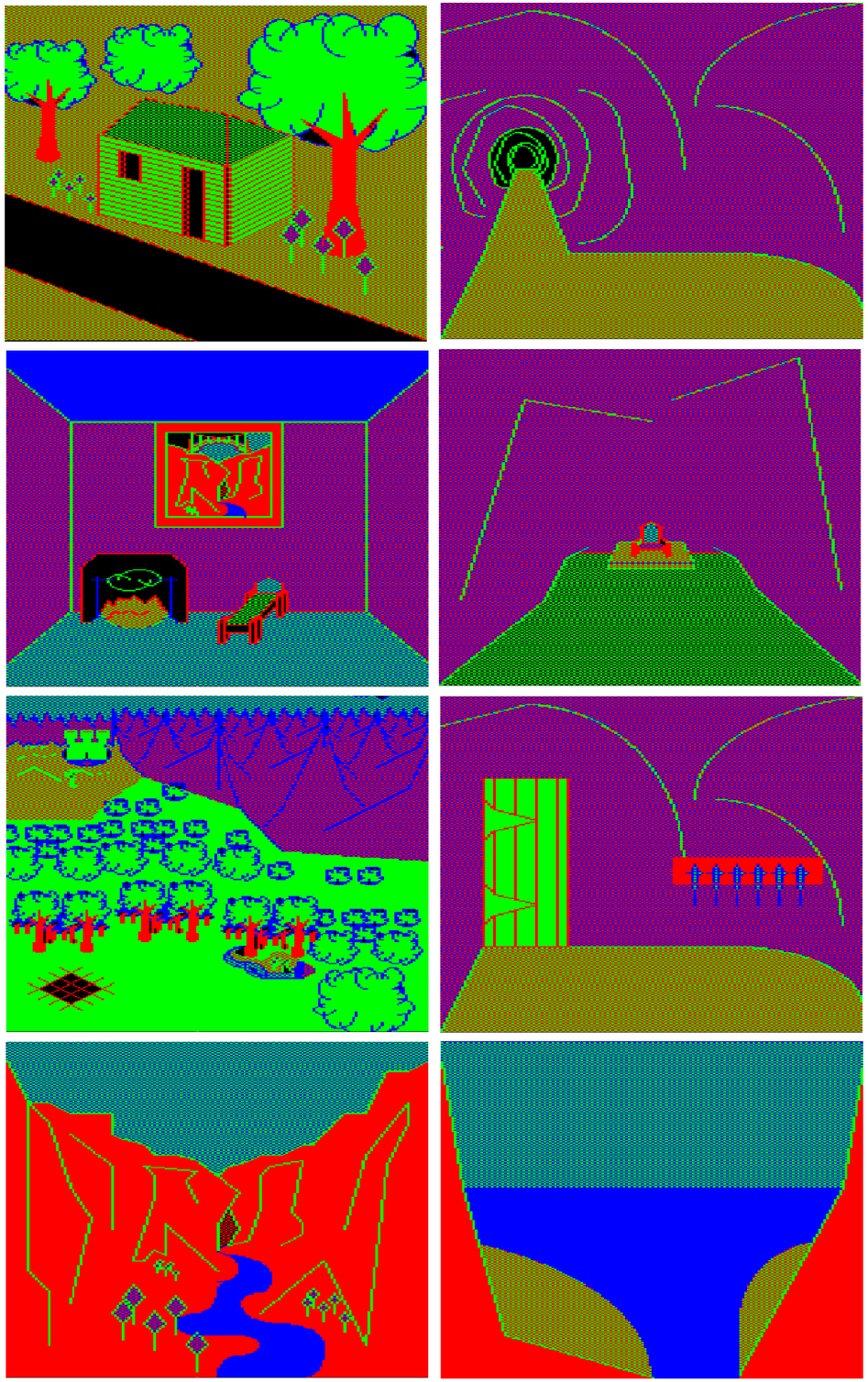
TWIN KINGDOM VALLEY

You needed your wits about you and a sharp sword handy in **Trevor Hall's** *Twin Kingdom Valley*, a game revered 40 years later as one of the best text adventures of all time.

Format: Acorn Electron, BBC Micro, Commodore 16/64 and ZX Spectrum
Publisher: Bug-Byte
Developer: Trevor Hall
Release Date: 1983

In May 1980, Tony Milner and Tony Baden were at Oxford University reading chemistry. They were fascinated by home computers and dabbled with writing software for Sinclair's pioneering ZX80 micro. "We were getting fed up with it," explained an exasperated Tony Baden to trade weekly Computer and Software Retailing magazine in October 1984. "There was no software [...] being sold at the time, [...] so we decided to start marketing our games and [...] made the move from Oxford up to Liverpool."

Armed with £10, they purchased 40 blank cassettes and peddled their 1K game compilations using £4.50 worth of advertising in Computing Today magazine. To their astonishment and delight, all 40 copies were sold within two weeks and over a celebratory pint in a local pub they dispatched the registration fee required to incorporate their new enterprise. The beautifully named Bug-Byte Software company frantically re-programmed its range to support the newly released ZX81. In early 1981 they moved out of Tony's bedroom and



into offices at Mulberry House, Canning Place, just around the corner from Bruce Everiss's Microdigital shop located on Dale Street.

Liverpool was a city like Sheffield and many others, suffering from social and industrial upheaval as it transitioned from its declining heavy industry in a shift towards a service-oriented economy. The rise of inexpensive home computers, the revolutionary BBC Computer Literacy Project and the fascinating with technology by children meant that shops such as Everiss's became a hub for self-taught, teenage bedroom programmers.

It was fertile recruiting ground for fledgling computer game companies such as Bug-Byte. They identified and employed the talents of David Lawson, Mark Butler and Eugene Evans who developed their first commercial hits including *Spectral Invaders* and *Spacewarp*. Sales were phenomenal, and Baden boasted of shipping units in the hundreds of thousands. By 1982 Your Computer reported that their workforce totalled 12 employees, all with an average age of just 19. By 1984, that number was 15 and the future looked increasingly bright for the Liverpool outfit.

Chris Cannon had been hanging around in the Computer Centre outlet in St. Johns Precinct in Liverpool. He joined Bug-Byte after his friend Eugene Evans invited him along to test a new Color Genie computer they had just purchased. Cannon created the text adventure *The Castle*, before deciding that he hated coding and moved into an administrative role within the firm, famously recruiting Matthew Smith and delivering the iconic *Manic Miner*. Part of Chris' duties was to evaluate the games that were being sent by the public, usually delivered on cassette in an envelope through the post. "[We] were getting 20 or 30 games submitted some days and most of them were useless and went straight in the bin" Chris told Paul Drury of Retro Gamer. However, "some were excellent" he recalled, citing games sent for the BBC Micro by Trevor Hall.

Hall was studying Computer Science at Manchester University and playing arcade games in the student's union in the evenings and between lectures. He forked out for an Acorn Atom, then a BBC Micro and sent some of his creations to Bug-Byte hoping for a commercial release. He told Jacob Gunness that "I think Bug-Byte was the first I tried, and they offered a good royalty rate." He was snapped up and contributed a considerable percentage of BBC Micro output for the Liverpool based publisher, including, *City Defence*, *Oblivion*, *Sea Lord*, *Dog Fight*, *Space Invaders* and *Galaxy Wars*. His final game and magnum opus, a text adventure called *Twin Kingdom Valley* arrived on the Beeb and Acorn Electron home computers in late 1983.

[Chris Cannon] When *Twin Kingdom Valley* arrived for review, it arrived like any other submission, but it didn't take long to realise that it was on a completely different level compared to the usual fare. From what I recall, when it first arrived, the game was already in a very advanced state. Unlike many submissions that were essentially 'proofs of concept' or buggy prototypes, Trevor's game was remarkably polished. It didn't need much work in terms of fixing or tweaking the core logic. Trev was a very quiet, unassuming guy - quite different from the 'rockstar' personas that some of the other coders were starting to gain at the time. He just got on with the work. He was incredibly focused on what he was doing.

Bug-Byte hailed its launch, say it was the "dawn of a new era in graphic adventures." The game's namesake valley was home to two feuding kings, hailing from the Forest and the Desert, and housed all manner of fairy tale creatures including dragons, dwarves and giants. At its heart was a treasure quest summed up simply in the game's instructions; "collect as much treasure as you can without being killed."

Valley's inspiration came from the many hours Trevor spent with friends playing the original *Adventure* on the Computer Science Department's mainframe computer. Wondering how adventure

---

[Left] Trevor Hall created an exceptionally flexible graphics language for *Twin Kingdom Valley*. Notice the use of repeated tree shapes for forest and cabin picture, and the use of scale, redrawing the canyon but at a small size in the cabin picture.

EAT YOUR HEART OUT, BILBO!

The Hobbit beat *Twin Kingdom Valley* to market, but drew the obvious comparisons on release. Keith Campbell in C&VG magazine contrasted the implementation of their non-playable characters.

Bug Byte, keen for those who had played Melbourne House's classic to buy their game too, ran a series of adverts with a cheeky "EAT YOUR HEART OUT, BILBO!" slogan.

games worked, he would sit around afterwards sketching out how the game's logic would work for a home computer system on pieces of green paper.

"Crowther and Wood's *Adventure* was legendary" Trevor explained in Acorn: A World in Pixels. "It seemed very easy to describe, so some friends and I sketched out a map and then I postulated how trivial it would be to describe a location on the map, and to connections to other places in a few bytes".

TKV's Adventure influences are obvious from the outset, mirroring the opening to Crowther and Woods' classic. The player in *Twin Kingdom Valley* starts on a road, running from east to west, with a small wooden cabin they've rented from the innkeeper of "The Sword Inn" nearby.

[Trevor] I wasn't interested in [using] other people's work. What would I then be making? The same game as someone else? The "Cabin on the road" [was] similar to the original *Adventure* to say "look, this is an adventure game", everything else was completely new work.

Despite Hall's insistence, the treasure quest also drew on influence from *Adventure*. Each "ill-gotten gain", gold, silver or diamond had to be "stashed" in the cabin, with the aim to score the maximum of 1024 points.

[Trevor] I never got very far in *Adventure*. There was some version on some system at the university, which we found fun to play, but time on those systems was limited, and we had to study sometimes. I think one or more of us had commented about the characters "standing still", [which meant] exactly the same puzzle to solve each time you played.

That predictability was something that Hall knew he had to address. He started designing a hybrid text adventure and used his experiences taking part in role playing games to introduce a basic fighting system. He added "strength" and "health" that were attributes of the player's character that could be affected by combat, hunger and thirst. Any lost points could be replenished by resting, eating and drinking.

It was an essential game feature for Hall, who expressed that the implementation of dynamism was an interesting exercise and that "programming is fun". He told Edge Magazine in an interview in 2006 "I'd seen adventure games on home computers before and they seemed really boring. The troll was always standing by the bridge. It made no sense! So I thought, why not make every creature in the game, including oneself, pretty much the same. Make everyone able to walk around, pick things up and fight. Let them interact."

[Trevor] I wanted more interaction. I guess I liked the creatures a little better than anything else. I connected with my creatures which everyone has some acronym for that I had never heard of at the time. I wanted them to be as real as possible.

They were very sophisticated ideas for the time, especially the

concept of pseudo-intelligent characters – those that could freely move around the game world, making decisions, fighting each other, taking objects and affecting the player's experience of the adventure. It was very reminiscent of the thought processes behind the development of *The Hobbit* – coincidentally both in development at the same time, born with the same frustrations with *Adventure*, but on different sides of the planet. The in-game instructions allude to Hall's thinking: "As you may have realised, this is not only the name of a game, but also of a mythical place in which two kings live. [...] This game is a simulation of the life in that valley. Whilst you play, other creatures will live their natural lives in the valley."

As Veronika Megler, on behalf of Melbourne House, wrestled with wizards and trolls in Tolkien's world, Hall implemented a series of simple rules and attributes that he applied to characters in the Valley. In an interview with The Classic Adventures Solution Archive [CASA], he told Jacob Gunness "I wanted creatures to make intelligent choices, to know when to walk and when to run. When they encounter you, they'll decide whether to fight based on how strong they are, or if anyone else is there too. If they're half dead, they'll make a run for it."

[Trevor] The questions were, how can you make it "fair", so that the same rules apply to "you" and "them". [There were no] special rules for [the player] in battles, [but] the tricky bit was in a turn-based game, who throws the first punch? This was balanced by the rule, you can fight first, but if you choose not to fight, to run away, everyone else gets their turn in before you can run.

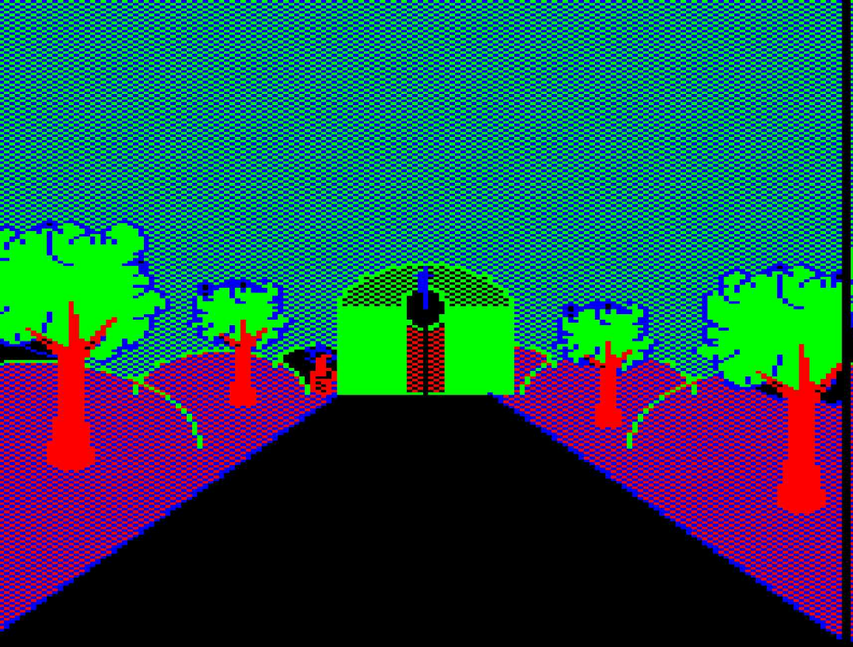
Trevor's interview with Gunness for CASA is fascinating. In it he details the exact configuration of the internal tables or databases that defined the locations, graphics and specifically the dynamic characteristic attributes that could apply to objects and creatures. It made for some "life-like" actions, and the illusion of intelligence achieved with simple decisions for the characters: Could they pick up a weapon, were they strong enough to fight, if not should they flee, or should they wander around instead? You could stumble across two characters fighting each other and then steal their misplaced weapons, or the non-playable characters [NPCs] would steal your weapon, or even run off with a piece of inventory or useful object. It was great stuff, and the kind of behaviour that elevated the experience beyond that of a bog-standard adventure game.

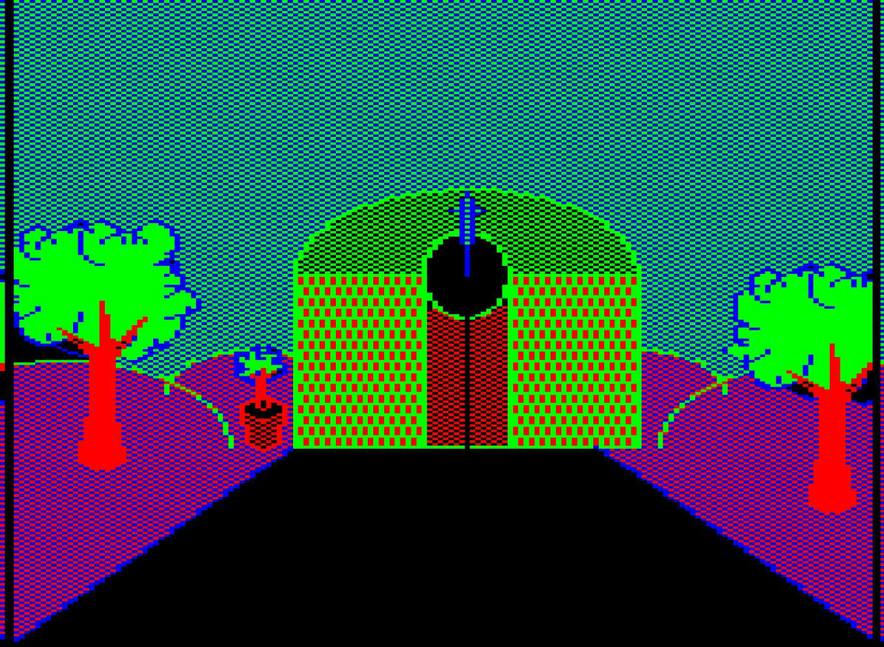
[Chris] At a time when most adventure game characters just stood in a room waiting for you to GIVE them an item, Trevor's characters had their own agendas. They moved around the map independently and interacted with the world in a way that felt truly groundbreaking. It was the first time I had seen a 'living' world in a text adventure. You could be in the middle of a puzzle and a NPC would just wander into the location, do something, and leave.

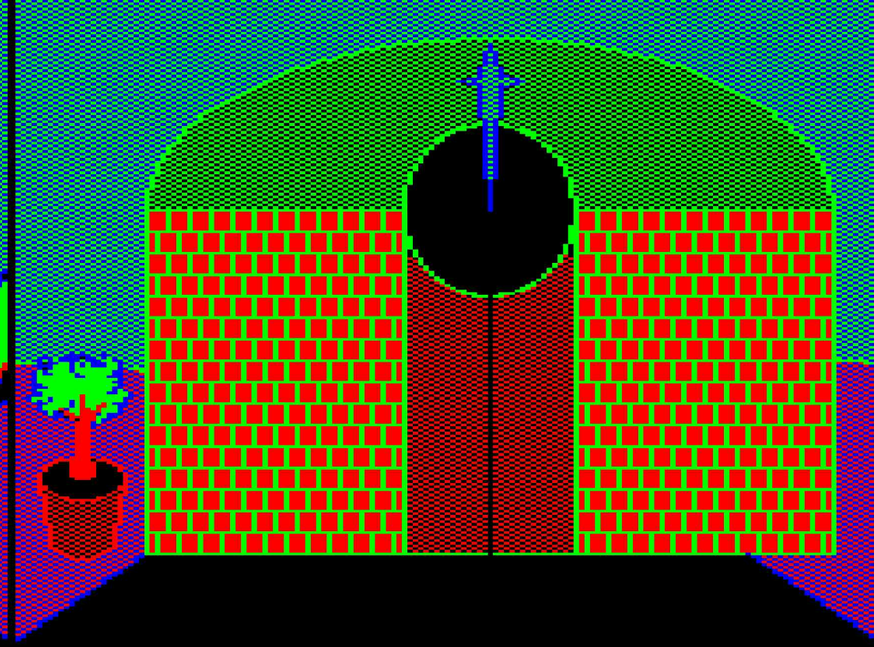
Adventure critic Merlin, scribbling for the popular long-running Electron User adventure column delighted on the NPC behaviour: "You will meet various characters most of whom, if armed, will attack you on sight. An elf will carry things for you but is sometimes reluctant to let you have them back. This same elf is also a positive nuisance when mapping underground - he kept picking up the objects I was dropping in order to make my map!" Hall told Edge magazine "I wanted creatures to make intelligent choices - to know when to walk and when to run. When they encounter you, they'll make a decision whether to fight based on how strong they are, or if anyone else is there too. If they're half dead they'll make a run for it."

This deviousness realised Hall's wish that there was a good chance that no two games would play out exactly the same way. Throw into the mix an overlooked collection of attributes that Hall implemented, a feeling of wellbeing in characters, a sense of self-awareness and one of loyalty. The NPCs could regard the player as friend or foe. Edge magazine covered one of Hall's favourite set pieces deep in the castle dungeon where freeing incarcerated characters made them grateful: "Ingratiate yourself with the giant by curing his illness and he'll join you in battle. Even better, storm the dungeon, hand out your spare weapons to grateful elves and suddenly, you've got a tooled-up posse on your side."

Health was a constant factor that had to be managed by the player.







[Above and Opposite] The picture is your view east. Trevor's ingenious scalable vector graphics language allowed objects, like The Sword Inn here, to be designed once and then reused as a subroutine, scaled down to appear in the distance or enlarged as the player approached.

Just moving around the map depleted energy, and by losing strength [or drinking too much beer] the player would feel weak and be more likely to die in combat. WAIT-ing around would replenish strength, but the Valley's mystical oasis Watersmeet would offer a more refreshing result if you took a swim.

There was a choice of weapon when fighting a foe – weapons could be rendered useless or were less effective against a certain type of adversary. A final variable of "fragility" was added to each object, meaning an over-used object would eventually break. All of it was very simple, but the rules were able to generate perceived complex behaviour, and it became a good early example of what would become "emergent" artificially intelligent behaviour. Hall saw this as a "world made up of equals" where NPCs were driven by the same motivations as the player and took their own "turns" in the game.

[Trevor] We played the game 100s of times, making sure that you could not easily ignore all the weapons and just go for treasure. So, it was balanced by playing over, and over, and over - adjust "Giant's maximum health" or "weight of a sword" until it all made sense. The characters had, mostly, not that much to do with completing the game, but became more a level of difficulty.

[Chris Cannon] I couldn't tell you the intricacies of any part of the game now, but I spent many an hour trying to break it. While Trevor had obviously put in the work to ensure the logic was sound, my role meant I was tasked with finding those hidden logic-bombs to crash it. I sat there for hours trying every 'wacky' command or illogical sequence of moves I could think of to see if the game would fall over.

The puzzles may not be the most convoluted and at times the game gift-wrapped you the solution to specific problems. Matthew Pennell a love letter to the game in his Return To *Twin Kingdom Valley* blog post mused over the blatant signposting: "Trying to either light the fire or sit on the bed triggers the same response: You don't have a flint." Its parser was limited [just 33 words in fact] and they were reassuringly included in the opening instructions. It had a short and functional *Adventure-like* approach to textual descriptions of locations. There was a lot [a lot!] of wandering around, back and forth, fighting, running, turning the lamp on and off, and opening bronze doors with bronze keys, but it was hugely compelling and atmospheric. Once the majority of the locations had been created, Hall played through what he had, and his thoughts turned to further enhancing the experience.

[Trevor] After just a little experimentation with "you are on a road", [I thought] how can I describe a location better than this? A few things happened. Locations point to other locations, via an "exit" so you can make more words by reusing the names of other locations, [and I wanted] the game [to] look 3D, so you can see distant locations.

Memory was at a premium, so to address the need for graphics, Hall created one of the first portable graphics languages for generating images. During his computer science studies at Manchester University he had experience of coding line and circle drawing algorithms. Using that knowledge he devised a system of arcs, lines, flood fills and scaled point plotting – the relative positions of line start and end locations on screen. He told Personal Computer Games magazine [who commented that Hall was "unwilling to give away any secrets] "it's a very crafty programming system" that could easily be implemented on other micros, having the relative positioning adjusted for the target screen resolution and colour palette.

[Trevor] Graphics [came] pretty early on in the concept. In a 32K BBC micro, I think 10K is [taken by the] screen, [and] allowing 1K or 2K for working memory, all games [would] have to fit into about 20K. The graphics took up a few K, maybe 4K to 6K, maybe more with the drawing code. I had no other idea how to make it fit.

Having a series of small, compact instructions on how to build images proved to be a very flexible solution. The use of relative plot points meant that each draw element (a line, or circle or whatever) could be moved around and started anywhere on screen and scaled up or down. The castle or waterfall for example could easily be changed to get bigger as you moved towards them, or smaller as you journeyed away. It was a nice compromise over the original true 3D that he wanted in the game.

[Trevor] I wanted to draw "what was ahead". I also loved recursion. after all I had previous written a chess game just for fun. So, the idea of using that to "make a forest" was just fun. Anyway, it saves a lot of text if you can see the castle getting closer.

Location artwork, provided by a friend, Pete Skinner was traced and optimised to fit within the constraints of the drawing routines. It took a little work to make it ready for the BBC, and Hall had to augment the overlapping of objects to minimise any unwanted behaviour from the flood fill routines. Finally, being able to call a series of instructions as looping "sub-routines" meant that certain images, or parts of images could be drawn over and over again, providing further extensibility, and an easy way to fill the screen with trees in a forest!

[Trevor] That wasn't much innovation to me. All languages, from the simplest assembler have "JSR" for "Jump to subroutine", so naturally, a language has subroutines, loops etc. The idea that something can be in the distance required scaling, so I just combined the concept "draw some other image, optionally scaled down by factors of 2."

"Saving a lot of text" wasn't the only novel aspect of the imagery. The location visuals provided clues for the adventurer. Despite not being mentioned in the description, in the opening wooden cabinet location there is a picture hanging on the wall. With closer inspection, more detail is revealed and it showed a bridge spanning a river, as it enters a cave between two high walls. It was a hint to what would happen later in the game with the introduction of a magic wand. Oh, and they could be turned off – perhaps something to placate the purists (and there were many of them) that text adventures, meant text adventures.

[Trevor] Well, that one was easy. "Canon picture with bridge" was just something like "location image 150", as that exact picture was used as a location image when the bridge has been created. So, I thought it would be fun to show it, as if it was "always there a long time ago".
Hall had never really planned *Twin Kingdom Valley* to be a product, and his commitment ebbed and flowed between studies and socialising. He was enjoying student life in his early 20s, so worked on the game on and off for between six and nine months. It was released at Christmas in 1983 where its full colour game inlay [sharing an art style similar to its sister adventure *The Castle*] benefitted from the influence of marketing consultant Bruce Everiss. Before his departure from Bug-Byte, he'd insisted upon professional presentation that mimicked popular culture, designed to lure in youthful customers.

The impact of a genuine graphic adventure in 1983 could not be understated. Bug-Byte rightly trumpeted this in all its press material, and the number of images was a boastful metric. *Twin Kingdom Valley* adverts proclaimed that the game had "over 175 screens". An adventure with full-screen graphics on the BBC and Electron was even more astonishing. David Elliot's conversion of his Nascom game *Eldorado Gold* [released by Program Power] and Bug-Byte's own *Dragon Quest* had reached the market earlier, but with exceptionally rudimentary visuals. Micro Adventurer magazine pushed the virtues of the game, saying "the picture-making capabilities of the computer are exploited excellently in most drawings",concluding that the quality of the graphics were "excellent." Bob Redrup called the style of game "illustrated adventures" over his love of the purest form, and in his Adventure Gamer's Manual remembered the game's release as a defining moment. "The first one of these that I ever saw was for the BBC Microcomputer - *Twin Kingdom Valley*," he extolled, "this was something of a landmark in its day. Each location was given its own illustration, and even though quite a number of them in fact had the same illustration, this was quite cleverly disguised." Not everyone was impressed. Keith Campbell, writing as the Adventure Supremo for Computer & Video Games magazine in May 1984 offered a mixed assessment. While he found the game "quite interesting to play" and the response time "quick", said that the "novelty of colour graphics, except where used as an integral part of an adventure, soon wear off."

"We played the game 100s of times, making sure that you could not easily ignore all the weapons and just go for treasure."

CURRAH COMPLIMENTS

Despite having a voice like a malfunctioning Metal Mickey (one for the kids), the Currah µSpeech support included with *Twin Kingdom Valley* did yield Trevor a pleasant surprise.

Thanks to its speech synthesis a young blind child was able to play and enjoy the game with his father. Trevor recalls receiving the letter, "I got [it], the Dad was so happy. We got a lot of fan mail at the time, most of which wasn't so unique. So Nice."

Unfortunately he no longer has it, more than likely leaving it along with a huge box of code printouts when he left Britain to live in the States.

But, as with *The Hobbit*, it was the inclusion of Hall's ambition for creatures that lived their lives independently of the player gave the game energy and its long-lasting appeal. Creating a sense of a living world, with graphics, became an elusive goal of adventure writers and adventure creating utilities over the coming years. Interceptor Micros' *Heroes of Karn* and *Jewels of Babylon* [**see Issue 04**] pushed the boundaries of graphic adventures, but It wasn't until the release of *The Illustrator* by Gilsoft in 1985, *The Graphic Adventure Creator* by Incentive Software and *The Professional Adventure Writing System* [again from Gilsoft] in 1986 that tools capable of the feat were in the hands of independent developers.

TKV's first entry into the Acorn computer charts came in February 1984, when it appeared at number 6, just a few places below *The Hobbit*. The following month it retained its appeal, and in April 1984 sales figures taken from retail and wholesale data saw the game ascend to the coveted number 1 position. It was serious stuff, beating off the challenge of competition from BBC and Electron classics released by Visions, Program Power and Superior Software. Shifting as many units as it did resulted in it being ported to most home computers around, with Hall doing all the 6502 CPU conversions himself. Its svelteness and smartness meant that most of the versions included the graphics, apart from the text-only Commodore C16 [released in 1986] edition.

[Trevor] [My port] to the C16 proved that the game engine, excluding graphics was less than 16K. Pete [Skinner] was paid better than beer money for the [graphics], as was a Z80 hacker [for the] Spectrum port, as we needed it fast!

That Z80 hacker is credited on the Spectrum Computing website as Joey Headen who was a fellow Bug-Byte employee. Trevor had collaborated with him [alongside Rob Pheonix] on the strategic sci-fi game *Star Trader*. The 64K of RAM in the Commodore 64 provided Hall with a small playground of surplus memory and there was room for the adventure to breathe a little deeper. There's the appearance of animations in some location graphics as the fire flickered in the wooden cabin, squirrels climbed trees and water dripped from stalactites.

[Trevor] The C64 version made use of its small amounts of additional memory with small animations and extra rooms. Well, it had a sprite chip, so I thought to use it. There is a grating from the spiral staircase that takes you to a hut in the mountains, where you get a silver dagger to give to the witch in the castle. In the BBC version there is no silver dagger, so you need to kill the witch with the staff. On the Spectrum version, arriving ten months after the BBC/Electron and C64, the Currah Microspeech peripheral was supported, but was best ignored – unless you wanted to play the game with *every* *single* *word* read in a grating robotic voice.

Despite the blockbuster revenue from *Manic Miner* and *Twin Kingdom Valley*, Bug-Byte faulted as most of its high-profile developers and product (in the case of Matthew Smith and Eugene Evans) departed citing disputes with management over wellbeing and the failed payment of royalties. The floundering company couldn't respond to growing competition, and by August 1985 charts in magazines like Commodore User and Popular Computing Weekly no longer showed *TKV* occupying the top positions. Newer titles like *Elite*, *Beach-Head*, and *The Way of the Exploding Fist* from Firebird, US Gold and Melbourne House started to dominate. Unlike its high-profile Liverpool-based competitor Imagine, which imploded spectacularly with high-profile debts, Bug-Byte opted for voluntary liquidation in May 1985.

The company's assets were acquired by fellow British publisher Argus Press PLC and the Bug-Byte brand and game rights passed to their subsidiary Argus Press Software Ltd. Managing Director "Big" Ron Harris made the commitment that Bug-Byte would be returning as a budget label, and *TKV*, under the new branding and with underwhelming new artwork did hit shelves in late 1985 priced at a modest £2.99. Compilation supremos Beau Jolly added it to various editions of its *Five Star* series of collections and it received an obscure release in the US from Santa Ana's Tri-Micro Inc under their *Gameware* line of software. To provide American and Canadian markets increasing value for money, the Gameware release bundled Hall's adventure as a dual-adventure package on disk alongside Charles Lowry's inventive *Corom*.

Search eBay for it and you'll be rewarded with multitude of cassettes up for grabs – surely a sign of sales success. It was critically acclaimed, even in C64 and Spectrum circles where the story telling had been judged by some to have moved on in adventure land. Micro Adventurer commented "This adventure is sure to become a classic", with Crash saying it was "well worth exploring by any adventurer".

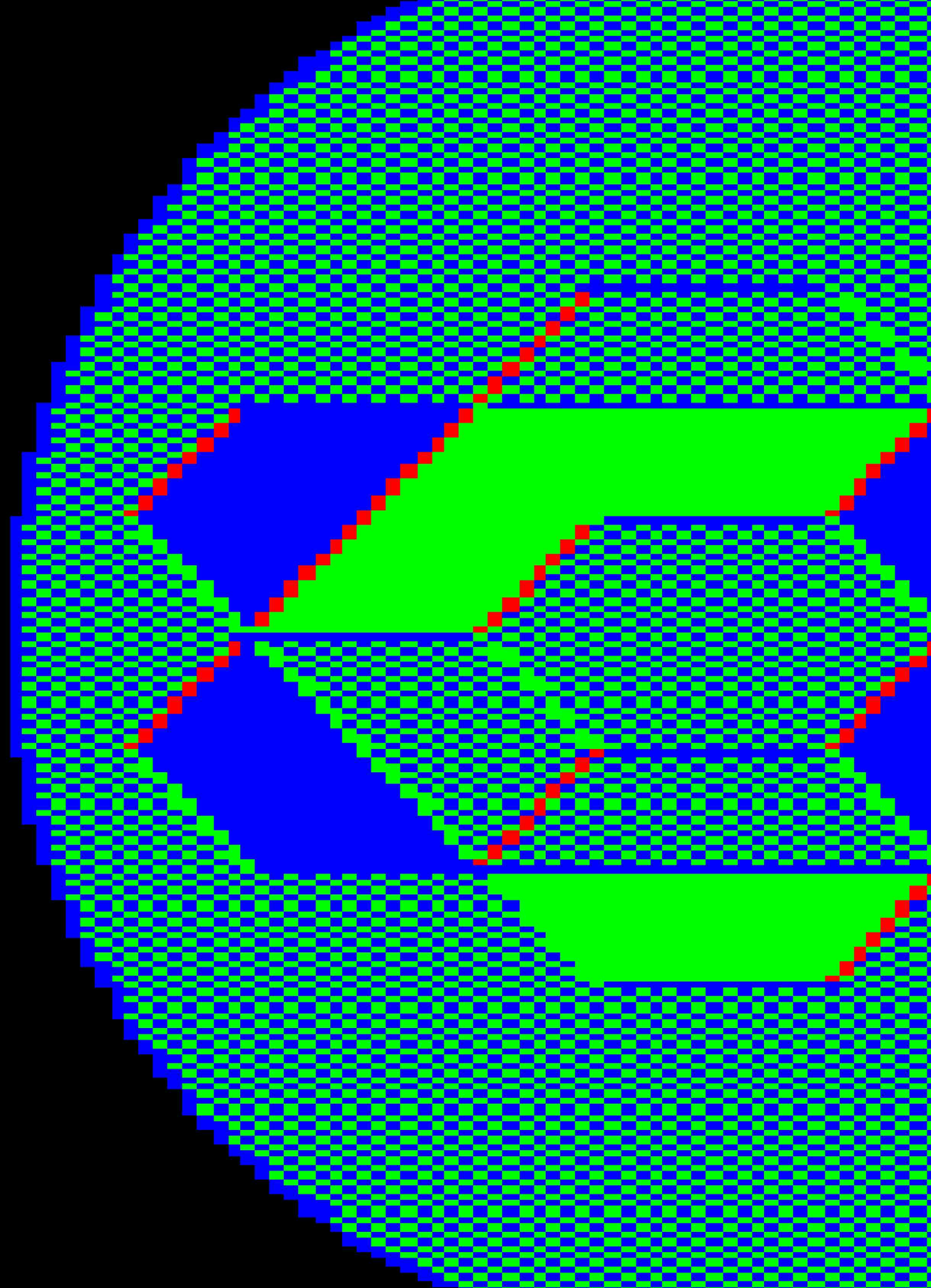
Ultimately, Twin Kingdom Valley is remembered not just for its "babbling brooks" and "hostile kings" but as an adventure worthy of being mentioned in the same breath as The Hobbit. It is testament to Hall's craft that he managed to cram such an ingenious adventure alongside the BBC's memory mashing MODE 2 graphics. From his various interviews it's obvious that he found it just as rewarding as a technical exercise as he did a creative one - akin to the Austins at Level 9 and the challenge of A-Code, or Mike Singleton and his Landscaping techniques. For British home computers with their limited RAM and cassette storage systems, programmers found a lot of enjoyment from trying to achieve as much as possible within the confines of that very tiny space.

Once you'd found all the bags of silver, hit the dragon with the staff and crossed the crystal bridge to steal the crown from the Desert King the game ended with the secret of life appearing in the room. What was the secret of life? After LOOKing at it you could let your own life ebb away as the machine painfully paid homage to Trevor's love of recursion and churned out a complex and never-ended diamond fractal on screen. Thankfully this painful process can be speeded up in a modern emulator and the game then ended with a rather cryptic message. It was a disappointing end, and a head scratcher to why this was included. The narrative defeating one of the kings and taking the treasure would have been sufficient. It's just one of TKV's foibles (along with the annoyance of having to reload if you die) but a forgivable one. For The Classic Adventurer, I won't forget its grandiose, its scale, ambition, and the graphics on my bloody Acorn Electron!

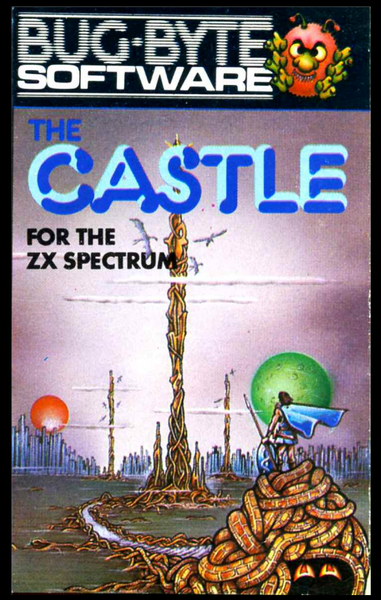
[Trevor] Yes, basically it was a treasure hunt game! There had to be a way to know that you are finished, and I wanted it to be a bit of a puzzle to find "the end". If I had 30GB of space, I would have made a ½ hour glorious cut scene. I had used up 99% of the memory already for the game, so needed some small way to let the player know he was done. [It was] purposely surreal, [and] my friends thought it was funny.

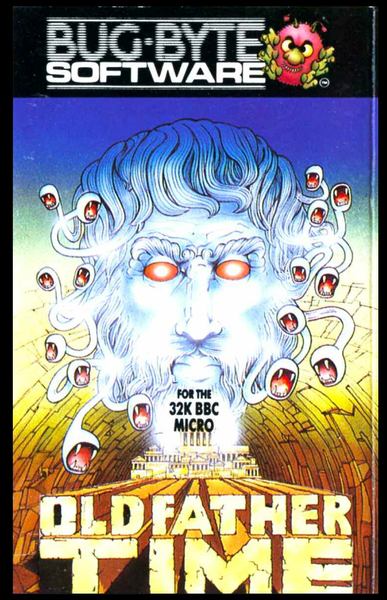
As for the message?

[Trevor] [I] needed [it] to fit in 100 bytes. THE SECRET OF LIFE IS



[Right] What was the secret of life in *Twin Kingdom Valley*?





[Above] Bug-Byte released a series of adventures for different micro computers, including *Adventure*, Chris Cannon's *The Castle*, *Old Father Time* and *SETI*.

SIMPLE YET COMPLEX, AS IS THE PICTURE OF IT. STAY COOL.

Sadly, we never did get to return to the Valley. There was a planned sequel called *The Valley of Gold* where the player had to search the land for the source of the River of Gold. "Near the end, you find a river of gold" Trevor recalled to Edge. "You have to wonder where it would go? And that's where the next game would take you. But how could you travel the river? What boat could withstand molten gold?" Hall shelved the project when Bug-Byte (a company he told Edge magazine was "truly a rotten outfit") collapsed owning him "oodles" of royalty payments.

[Trevor] I had started on a dynamic landscape tool such that forests, deserts, etc would be different in each game restart, but never wrote any game code. No story line was written, but you would be able to travel the river of gold to a vast land. No more pathetic 175 locations. I recall some ideas like "A wooden boat would burn in a river of molten gold", so the game had to be started well. Some part of the original game would lead you back to that river.

Adventure Coder magazine speculated that Hall's sequel would be a Commodore 128 game called *Valley of the Source*. Editor Chris Hester also claimed the game's dynamic landscape would be able to generate 500,000 locations. It would have a *Lords of Midnight*-esque engine, with the character able to look in every direction of the compass. Hall's interview with Edge prophesised his vision for the game would need a machine with at least 64K of RAM and be able to support creatures into the hundreds and locations into the thousands.

Trevor ceased game development leaving the industry soon afterwards. Following his exit, he left Britain and eventually moved to California, where he pursued a career in tech outside of the gaming sector for over two decades. In 2006 he teamed up with Ian Pare and his multimedia agency Silicon Magic to develop a mobile version of the original game. He provided his expertise in creating tools to manipulate the original databases and worked on the new graphics engine code.

Pare told Retro Gamer "I wanted to do a phone game you could play for a long or short time, put down and then come back to it where you left off. An adventure seemed to fit the bill. Twin Kingdom Valley had always stuck in my mind, partly because it was the first text adventure I'd played." After so long, and a change of country it wasn't a surprise that Trevor no longer had access to the original source code, so it was a mission for Silicon Magic's Pare to make use of modern technology, arm himself with a binary image of the C64 game and attempt to unscramble the code.

Ian told Jacob Guiness in 2006 that "armed with my copy of Rodnay Zaks' "Programming the 6502" and a dis-assembler, I reverse engineered the game to give me full 6502 source, which helped me understand how the game worked. Then it was a case of translating to Java and getting a prototype running on a phone."
The mobile port was released on the 10th of August 2006 on the Nokia 6230 and 6230i, where The Pocket Gamer website hailed its triumphant return. Reviewer Paul Drury, an avid adventure fan delighted in the effort: "Twin Kingdom Valley is a remarkable achievement in its own right", he said, "but its greatest triumph may be to introduce a whole new audience to a particularly rich and largely forgotten realm of video gaming." Paul concluded that "[*Twin Kingdom Valley* was] an ancient quest, beautifully reborn."

Trever was also impressed with Pare's work, and so enthused with the work-in-progress version ended up contributing new locations and updated graphics. Once the game launched he assisted in moderated the publisher's help forum, answering questions from fans. Despite the positive acclaim the game failed to sell, perhaps ahead of its time and technology, unable to be fully appreciated on the limited screen resolution and usability of the target mobile phones of that period. The Classic Adventurer has attempted to contact Pare via the now defunct Silicon Magic website to and obtain the Java source. A port to Chris Ainsley's superb Adventuron system would ensure the game could be enjoyed on modern systems.

Trever Hall himself has several incomplete versions that he's attempted to bring up to date to but has yet to find the time to finish one. With the Java version incompatible with modern browsers and mobile technology, for now, we'll have to be content with 8-bit emulation [and the blessed relief to avoid the instant deaths of being able to SAVE a snapshot at any position].

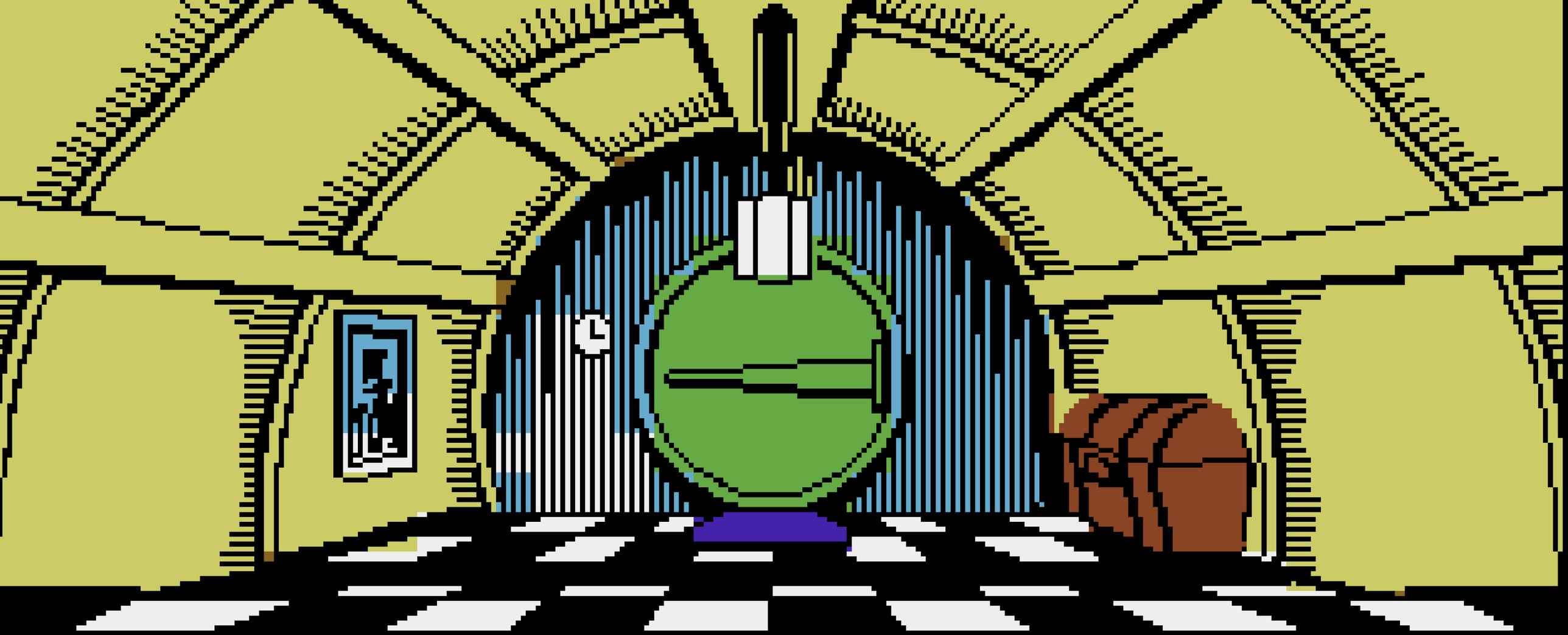
Let's hope we will all be sailing on molten gold, walking in Valley and hitting Dwarfs with hammers before too long.

DESERT ISLAND DUNGEONS

Having stashed his ill-gotten gains in a wooden hut, **Trevor Hall** sets sail from Watersmeet, only to be shipwrecked on a desert island far away from Twin Kingdom Valley.

I would sit there and write my sequel, *The Valley of Gold.*

I don't get that excited about other people's work. Most of the fun is creating?



EL HOBBIT

The Hobbit remains one of the most iconic and technically intriguing text adventures ever created, a landmark that still challenges and inspires developers decades later. The Classic Adventurer talks to **Antonio J Santos**, a Spanish developer about his ambitious effort to reimagine a classic using the *DAAD* multi-machine adventure game creation system.

Format: Commodore 64, Commodore Amiga, Atari ST
Developer: SrHead (Antonio J Santos)
Price: Name your own price
Release Date: April 2023
Website: https://srhead.itch.io/el-hobbit
DAAD Ready Telegram Group: https://t.me/daadready/1#

Where did your love of computers start?

[Antonio] My first computer, back in the 1980s, was a Sony MSX. I only had it for a few months, but that was enough to take my first steps in *BASIC*. Shortly afterwards, I had the opportunity to trade it for a Commodore 64, which I still own to this day. Over time, this was followed by the Amiga 500 and, later on, PCs.

What were your favourite games?

[Antonio] Adventure games have always held a strong appeal for me. In Spain, thanks to the company Aventuras AD, we were able to enjoy some outstanding text adventures. *La Aventura Original* and the *Ci-U-Than Trilogy*, which includes *La Diosa de Cozumel*, *Los Templos Sagrados*, and *Chichén Itzá*—were particularly important to me, as they were the ones I could enjoy in my own language. Beyond interactive fiction, I also have great memories of 8-bit titles such as *Head over Heels* and *Solomon's Key*. Later, in the 16-bit era, I particularly enjoyed all the graphic adventures from Lucasarts, as well as the *Eye of the Beholder* series by SSI.

How did your efforts to port *The Hobbit* start?

[Antonio] I discovered that *The Hobbit* was not just a simple adventure game, its author had attempted to bring the spirit of the book into the game itself. The descriptions of the locations closely mirror those in the original text, which I found deeply impressive. This motivated me even more, as I am a great fan of the work of JRR Tolkien. I knew that Tolkien himself had supervised translations of his works into other languages, so it was particularly disappointing for me to realise that I wouldn't be able to enjoy the game in my own language.

Did you still manage to play and enjoy the original game?

[Antonio] Yes, I later played it extensively, almost to exhaustion, always with the goal of uncovering its hidden mechanics and bringing them into my own translated version.

When did you have the idea to create your own port/translation? The original effort was long before the *DAAD* port?

[Antonio] Indeed, the original version was programmed in assembly language, which was beyond my reach at the time. As a first attempt, I tried to use tools such as *PAW* and *GAC*, but I quickly realised it was

not feasible: *The Hobbit* was far too large and complex for those kinds of utilities. So I decided to develop my own version for the Commodore Amiga, where I could use a high-level language without the usual memory constraints. I got to work and built a parser that, in terms of syntax, was quite similar to *PAW*, although it took me a considerable amount of time to complete. Eventually, the opportunity arose to use *DAAD*, the professional tool used by Aventuras AD. Unlike my own parser, *DAAD* gave me the ability to release the game on multiple platforms, which proved to be a decisive step forward for the project.

How long did the translation take? Was there any element that wouldn't translate from English to Spanish?

[Antonio] I remember spending a great deal of time ensuring the texts were translated accurately. The first step was to verify that the original descriptions matched the exact paragraphs from the English edition of the book, as Veronika Megler had mentioned in more than one interview. As a result, much of the translation process involved locating those same passages in the Spanish edition of the book and carefully substituting them. It was not an easy task, I had to reread the book many times, which required a significant investment of time. However, it was worth it as the recreation gradually took shape and became increasingly faithful to the original. At last, I could begin to truly enjoy this adventure in my own language.

Was it a case of playing the game repeatedly and documenting it?

[Antonio] Yes, exactly. I would load the original game into a Commodore 64 emulator and start playing. I pushed every location to its limits, trying almost impossible situations to understand how the game behaved. For example, I could command Thorin to pick up a goblin and then type PICK UP THORIN to see how the goblin would react. You can do some really bizarre things in this game. Translating all of that into code afterwards was a real headache.

Did that work for puzzles? Did you document responses from every VERB/NOUN combination, or did you look for puzzles that worked?

[Antonio] For each puzzle, I researched online and browsed fan forums to find all the possible solutions discovered by other players, while also trying to uncover my own. I believe I left no stone unturned. [...] It was essential to uncover the game's different endings, both successes and failures. The translation had to be an exact and faithful reflection of the original game.

Why did you choose *DAAD* and *DAAD Ready* for development?

[Antonio] I had experience with the CONDACTS in *PAW*, since I based my Amiga parser on this tool. Essentially, what I did was use them as a model and adapt them to my own implementation. *DAAD* was the professional tool used by Aventuras AD to develop their games, and for years it was the dream tool for anyone hoping to create their own text adventure. [...] However, the original system was difficult to handle, as it couldn't run directly on modern computers without using emulators like *DOSBox*.

So how did you manage to get around that?

[Antonio] This is where *DAAD Ready* comes in, developed by Carlos Sánchez. [It] gathers all the necessary tools and emulators to work comfortably on current machines, allowing the author to focus entirely on creating the game without worrying about technical details. *DAAD Ready* is simply excellent. It's designed so that all you have to do is program. By running a simple script it generates the executable and the final distribution and launches it automatically in the appropriate emulator—saving me a tremendous amount of time. Although the original intention of its creator was to make things easier for novice authors, experienced users can also take full advantage of it and harness its full potential.

How easy was it to code the PSI behaviour for *DAAD*, which isn't really designed to have the flexibility that *The Hobbit* engine did.

[Antonio] This was, without a doubt, the most demanding part of the entire development. Analysing the behavior of each character in the game took me countless hours playing the original Commodore 64 version, which in my opinion is the most advanced of them all. I believe I managed to reproduce around 98% of the behavior of the original game's PSIs. [...] Once the original algorithms was understood, replicating their logic was not particularly difficult. Moreover, the *DAAD* language offers great flexibility to create PSIs that are truly complex and exhibit very detailed behaviors.

How does your PSI handling work in *DAAD*?

[Antonio] Each PSI has its own process. For example, process 21 belongs to Gandalf and contains 461 lines of code on its own. The real challenge lay in the large number of PSIs each with its own distinct behavior pattern. This translated into an enormous amount of code and, consequently, high memory requirements. Add to that the space occupied by the *DAAD* interpreter itself, and technical limitations were unavoidable. Ultimately, it was impossible to create complete versions for machines like the MSX, Amstrad, or Spectrum due to memory constraints. Surprisingly, the Commodore 64 version could be completed, thanks to the capabilities of *DAAD Ready*, which allows the game images to be loaded directly from disk.

Did you drop anything from the original for *DAAD* to work?

[Antonio] In the Commodore 64 version, I simplified some aspects, although these involved very obscure situations that did not affect the game's progression, such as the behavior of certain PSIs when picked up or carried by others. In the original *The Hobbit*, to escape the dungeon, Thorin or Gandalf had to pick up the player to reach the window, as Bilbo was too short. In my C64 version, the ability for a PSI to pick up or carry another character was limited to only a few characters. [...] The possible combinations were countless, and coding all of this was a real headache that consumed a lot of memory. For this reason, it was simplified in the C64 version, while the Atari ST and Amiga versions retained the feature fully intact.

What other technical limitations did you have?

[Antonio] Another major memory consumer was the function of talking to PSIs. You could talk to all of them and even give them orders - though whether they obeyed was another matter. Additionally, each PSI responded to the player's requests in a different way. Then there were containers - objects that could hold other objects and even characters, such as the chest at the entrance or the barrels in the King's cellar. Furthermore, every PSI that died left a corpse, which automatically became a container object. If it was carrying any items, you could retrieve them.

Is it true that the technical limitations prevented a version for the Spectrum and other smaller RAM micros using *DAAD*?

[Antonio] Yes, creating versions for Spectrum, MSX, or Amstrad meant cutting features that the original game allowed. Approximately only 80% of the game could fit on these microcomputers. It's possible that in the future *DAAD* interpreters will be able to take advantage of the Spectrum's 128K. At that point, it would be feasible to develop a complete version for this machine.

You hinted on your itch.io page that a fully enhanced PC version might be on the way?

[Antonio] As for the enhanced PC version, another person was going to take care of its development. They helped me a lot as a beta tester and also contributed to improving the EAAD IDE for *DAAD Ready*, which became an indispensable tool. However, in the end, the project was put on hold, and I don't know if it will be resumed in the future.

Was there any temptation to produce a version that was more refined than the original, i.e. removing Thorin constantly singing about gold or making Gandalf more useful?

[Antonio] As for me, no. This was an entirely personal project, and my goal was always to make it faithful to the original, just as it was designed by the original programmers.

What about the notorious HobbitBugs? Did you maintain those – such as getting into the chest, or did you fix them?

[Antonio] They were reproduced exactly as in the original game. I looked for all the known bugs and recreated them, and even discovered some others.

Did you manage to include all of the locations?

[Antonio] All of them, absolutely. I recall that in *The Hobbit: The Software Adventure*, there were some locations that didn't appear in other editions, such as the "Deep Marsh," which even had its own graphic. Additionally, in this version, every location was associated with a graphic.

Given this opportunity to enhance the original. Did you give any thoughts on expanding the location count, or bringing in additional characters from the book, such as Tom Bombadil for example, that were missing in Melbourne House's original?

[Antonio] Certainly, on 16-bit computers, the game could be enhanced and almost anything could be added, since there are no memory or storage limitations. However, we would be talking about a different game: an improved remake. In my humble opinion, I think the game is very enjoyable as it is. It's highly replayable, since no two playthroughs are ever the same.

You've taken the C64 graphics and upscaled them. How did you do it and what were the challenges? Was it a case of using modern graphics tools to scale the pixels?

[Antonio] The graphics were created based on captures from the originals. Depending on the target machine, I followed one process or another. For example, for the Commodore 64 version, I processed the capture in *DPaint* on a Commodore Amiga 1200, loading the C64's 16-color palette and rearranging the colors. Then, using a utility called *Pixcen*, I converted the PNG format to ART format, readable by *DAAD Ready*, leaving the file ready to load into *DAAD*.

Did you have to modify any of the images for different machines?

[Antonio] I followed the recommendations in the *DAAD Ready* manual. I did have to modify some graphics. The originals were 256 pixels wide, but the screens of the C64, Amiga, and Atari ST are 320 pixels wide. I didn't like the black bars on the sides, so I extended the graphics to fill the entire screen.

Did you consider creating new graphics, or replacing some of the graphics with digitised images for the Amiga or ST?

[Antonio] It crossed my mind, and although I would have liked the 16-bit versions to feature their own enhanced graphics, I'm a terrible graphic artist. Therefore, I settled for adapting the graphics from the original 8-bit game.

So from start to finish, how long did the project take?

[Antonio] 26 months, as I can verify from the commits. There were many small iterations—up to 106—but also three major revisions, which primarily involved changes aimed at optimizing the code. It was a continuous process, with constant improvements at every stage.

What has been the response from the adventure community to the game's release?

[Antonio] According to itch.io data, the Spanish version has reached approximately 2,000 downloads, which has been quite a surprise for me. I did not expect the project to achieve such success, especially considering that it belongs to a very niche genre like text adventures and is targeted exclusively at the Spanish-speaking audience. Furthermore, we are now experiencing a thriving moment for this genre in Spain, with two highly active Telegram groups and the Club de Aventuras AD magazine published both in print and online and even two books released on the subject. It seems we are witnessing a true resurgence of text adventures.

Will there be an English version?

[Antonio] I had considered it, but it seemed pointless since the Spanish version is practically a replica of the English one. However, there are no versions of The Hobbit for Atari ST and Amiga [so] creating English versions would make sense.

Do you have any plans to port any other text adventure games or any others in the Lord of the Rings series to *DAAD*?

[Antonio] No, I've had enough with *The Hobbit*. Although I've started several projects to adapt some of the hits from Infocom and Level 9 for the Spanish-speaking audience. Tey are currently on hold. I don't rule out returning to them in the future.

Why do you think Spain has such a thriving text adventure community?

Antonio] Could it be our adventurous spirit? I honestly don't know. I think text adventures have something special that draws you in and captivates you. Interestingly, most of the people I know in this world not only enjoy playing them, but many are involved in creating them in one way or another. Programming adventures is as exciting as playing them. Furthermore, in Spain, adventures have always been somewhat of an unfinished chapter. By the time text or conversational adventures truly flourished, around 1989, it was already too late: the genre was declining in Europe, 8-bit machines were giving way to 16-bit systems, and the release of graphic adventures like *The Secret of Monkey Island* effectively sealed its fate. Perhaps the resurgence in recent years is a way of finally settling that old score.

Finally, if anyone wanted to create their own adventure using *DAAD*, would you have any recommendations on how to start?

[Antonio] *PAW* was known for its ease of use, so for anyone who used *PAW* in the past, *DAAD Ready* is its natural evolution. It continues to be updated and improved by its author, who is very active in this community. The best way to get started with this tool is through the very active English and Spanish Telegram group, which offers all kinds of tutorials and examples.

My thanks to the generosity of every contributor, who gave their time to humour me and answer questions they've been asked a thousand times before.

Special thanks to many friends, retro acquaintances, and text adventure geeks, including the usual kindness and help from:

Fergus McNeill
Tim Gilberts
Gareth Pitchford
Chris Ainsley

A non-exhaustive list of references and other useful information:

Books, Magazines and Fanzines

Retro Gamer Magazine, Future Publishing
Twilight Inventory, Gareth Pitchford
Spectrum of Adventure, Thomas A. Christie, Extremis Publishing

Documentaries

From Bedrooms to Billions, Gracious Films
The Hunt for 'The Hobbit's' Missing Hero, Great Big Story

Websites and Blogs

Spectrum Computing
8bitAG
The Digital Antiquarian
Internet Archive
The Classic Adventures Solution Archive
Renga in Blue
Level 9 Memorial

Research Papers

There and Back Again: A Case History of Writing The Hobbit, Veronika Megler
Digital Gaming and Tolkien, 1976 - 2015, Helen Young

Artwork

Red Moon
Roger Kean, Artwork of Oliver Frey

The Hobbit
Melbourne House, 1982 & 1985

Nick and Peter Austin of Level 9
Sinclair User, EMAP Business & Computer Publications, May 1985

Scapeghost
Level 9 Computing, 1989

Twin Kingdom Valley
Bug Byte Software, 1983

Adventure I
Abersoft, 1982

The Classic Adventurer
Written and designed by Mark James Hardisty

About the author

Mark James Hardisty is from Sheffield. His weekly pilgrimage to Just Micro as a child left him with an indelible love for Gremlin Graphics.

You can find Mark at @hardistymark, where he tweets about games, getting kids coding, The Cannonball Run, and his favourite game - *Elite* on the Acorn Electron.

This work is dedicated to:

My wonderful family – my mum Val, my beautiful wife Helen, and daughters Amelia Rose and Kitty Mae.

Fergus McNeill, a genius, and one of the kindest and humblest people I have had the pleasure of meeting. Thank you for *The Big Sleaze*.