

> Remember

By Hugo Labrande

Issue #17 : A little bit on Adventuron

I have had the pleasure of interviewing Chris Ainsley, the creator of the Adventuron system, which has grown in popularity in recent years, and even won the XYZZY Award for Best Technological Development in 2021! This is a great, honest, open-hearted interview that retraces the Adventuron journey and talks about Chris's vision for his tool, and its future. I am delighted to have this in >REMEMBER; thank you very much, Chris!

Hello Chris! Could you introduce yourself to our readers?

<< Waffle Alert >>

About myself? How to choose a public face? Well, I'm British, English, in my mid-forties (how it happened I do not know). I'm a manager in a financial company and a model-based development advocate. I hail from Thornaby in North Yorkshire. I love old movies - especially old British movies and recently old Japanese movies. Not all old movies of course, not the rubbish ones, unless they are so rubbish as to pass into good.

Childhood. I had a fairly typical childhood (in the golden age of being a child). I had a comfortable childhood and although my parents were not wealthy I was surrounded by love. It was a good time and place to grow up - even though at the time I longed for the California lifestyle that was everywhere in 80s TV and movies. You always long for the other. The other you long for in the future is the past. I'd ride my bike to my friend's house. We'd also go exploring - going further than was allowed on the bike. Knowing to come back for tea-time. Every day was an adventure. Well, perhaps not Sundays. My family lived in a relatively new estate (in the 80s). The houses were not old, and unlike modern builds today, the roads were wide and designed for living rather than maximising profit. It was a child explorers paradise. There were so many nooks and crannies. So many places to discover.

My first computer was a ZX Spectrum 48K, which was a shared computer between myself and my older brother. I started coding early on, with lots of type-in BASIC programs. My dad's friend "Steve" came over a few times and showed my brother and myself how to type in the programs from magazines and what all the strange words on the screen meant. He also brought over games such as Lords of Midnight, Hall of the Things, 3D Ant Attack, and old Crash magazines. The memories are hazy but the feelings are crystal clear. It was exciting. I instantly knew I wanted to become a programmer and boringly, and as predicted a decade earlier, I went on to study Software Engineering at University, doing quite well in the end.

Anyway, at university my area of interest for my final year project was metaprogramming. The art of high level abstractions. Hmmm, that might come in useful someday :-). What I really wanted to study at university was parsers and compilers, but through a fluke of scheduling, the software engineering students were not offered the module, only Computer Science students got to study it. Drat. I'll have to teach myself then. That might come in useful someday :-).

Eventually went into pharmaceutical IT - working in "validation", conducting code reviews for safety, then finance IT - where I eventually became a software architect for a back office job processing system (amongst many other tasks), then into entrepreneurship (which is technically now a side-gig for weekends), then very recently back into finance IT (putting my entrepreneurship on the backburner). I still code, but I'm not primarily coding in my day job, although I do get my hands dirty from time to time. In my spare time I still code a lot although perhaps I've chosen to specialise in a very niche area which worries me. I try to keep my eyes open.

During the past 5 years Adventuron was born, but more on that later.

Would you mind talking a little bit about what text adventures mean to you?

My two earliest text adventure games were *The Hobbit* (by Melbourne House) and *Smugglers Cove* (by Quicksilver) - both on tape for the ZX Spectrum. Both were illustrated text adventure games. My earliest memories of text adventuring were with the Spectrum hooked up to the family TV, trying to figure out how to escape the dungeon in *The Hobbit*, with the whole family battling with the parser. *Smugglers Cove* was a game I think set in some caves. The parser looked like it was written in BASIC, the graphics were primitive, and there was basically no colour at all. It was blue and black monochrome (or so I remember). I remember struggling with the

game and I didn't get very far, but I did enjoy trying hundreds of different verb and noun combinations. I guess I didn't have much common sense or world experience at that age to play well.

I enjoyed all kinds of games, and text adventures were not particularly a beloved genre but rather one of a number of genres that I loved - but there were many games where 11 year old me couldn't achieve anything at all. I recall *A Harvesting Moon* being a game I really struggled to get anywhere with and had this feeling of being barren and spooky. I tended to gravitate towards games with more of a puzzle focus, and with graphics, but I really enjoyed trying to figure out how to change the state of the games with verb nouns. The Adventure International games were really simple but really addictive. I loved *Gremlins*, *Robin of Sherwood*, and *Rebel Planet*. I played text adventures into the twilight years of the Spectrum, often ordering games from Zenobi, and I know there is a big stash of Zenobi titles (such as *The Beast*, *A Legacy For Alaric*, *The Balrog And the Cat*) in my parents' loft. I had a bias towards big bright colourful text which is a mainstay of games developed with THE QUILL and PAW by Gilsoft. I guess I'm shallow that way.

Favourite authors are Linda Doughty (who wrote the Beast), Fergus McNeill for his funny and oh so well themed Delta 4 spoofs, anything by Scott Adams or Brian Howarth, games by John Wilson or one of a number of Zenobi stable authors, and Colin Jordan - who wrote the excellent and criminally neglected "The Famous Five" which is the most best book adaption I can possibly imagine.

Around 1993 I became an Amiga owner and other than a brief introduction to Infocom via *The Lurking Horror*, I kind of didn't think about text adventures very much during this time. I had moved into graphic adventures and the supply of illustrated text adventures, to the best of my knowledge at the time, which was not very good actually, had disappeared. I am now aware that both the interactive fiction and text adventure scenes (which I view as completely different) were ongoing. Zenobi was still selling games over a good chunk of that period but I wasn't really in the community at all for the next 15 years.

<< Time passes >>

Around 2009 I became aware of Inform - always intending to write something with it, but never seemed to have the time or the talent - to make a game in the narrative style. I understood that Infocom's style came to define the community via tools like

Inform, and text adventures were now being called interactive fiction, and games were now being called stories. Inform was and is excellent but I personally failed to connect with that feeling I had when loading up the games that I loved in the 80s. Colourful text and/or colourful graphics, lots of theme customisation, simple parser, puzzle boxes. Being a sufferer of aphantasia, pure text games were mostly not my thing and I really didn't like the way the text flowed in the very common z machine engines.

Perhaps weirdly, I like a bit of a delay between submitting a command and the response appearing. Engines such as parchment and Frotz seemed to not really think about text delivery flow being part of the experience. It seemed that they treated delayed text flow as an issue that was rectified with modern hardware, a bug to be fixed. If you press ENTER and the text appears within a frame, then to the human mind, it looks like the game didn't need your input - it knew you were going to type that. Your finger isn't even fresh off the ENTER button and the parser is already telling you what another character said in response to the thing you just said - but your brain knows they never even had time to hear what you said. It's like the character knew what you were going to do before you did it. It's very unsatisfying and I don't like it. I want an engine to consider my command then execute it. Tiny things like that bother me..That said, anything subjective like this is configurable in Adventuron. Authors decide and players can override. Every game in Adventuron for example allows users to override the font and deliver flow. The author controls all the defaults but the player decides if they agree or not. It is perhaps an accident of evolution that parsers were slow for quite some time, and we were able to experience the joys of the goldilocks zone of response time. Too fast (such as frotz) - it's reading your mind, too slow (as in *Lord of the Rings* for the ZX Spectrum), it's frustrating to play.

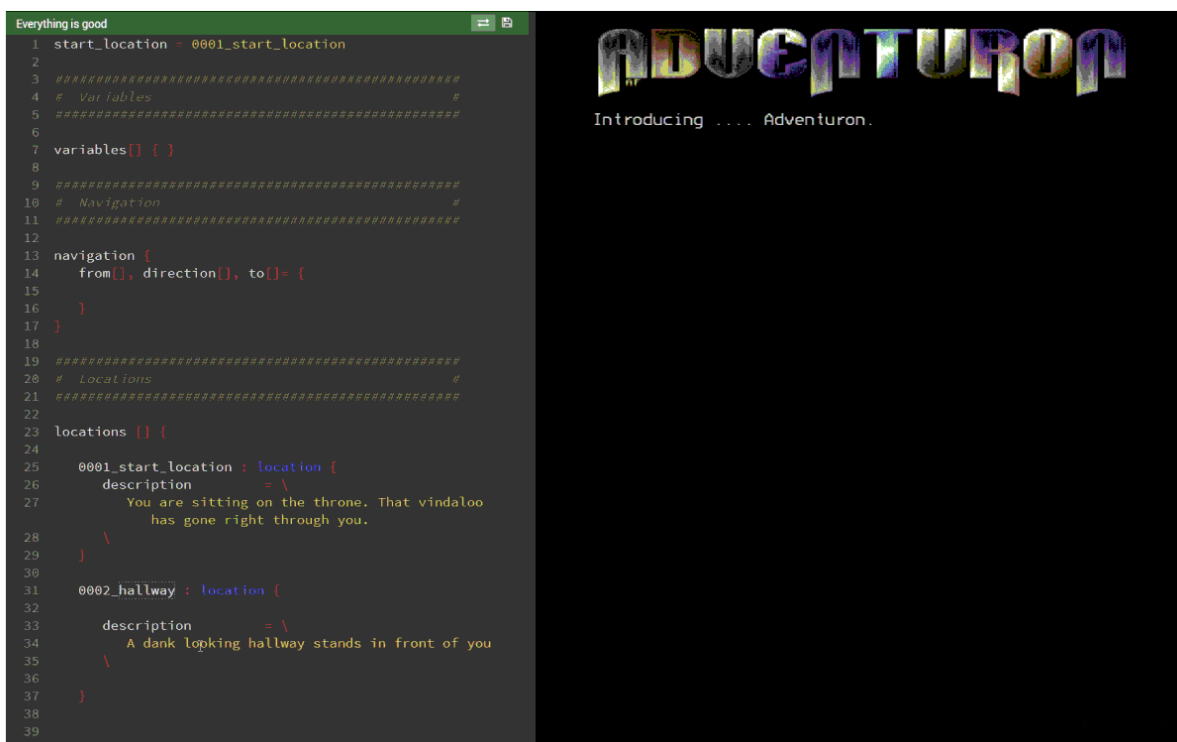
You are the creator of Adventuron, a great utility to create text adventures, that recently won the XYZZY Award for best technological development. Can you take us back to the very beginnings of the project? What was the idea? The core design principles?

Adventuron started as a technical demo for a language technology I'd been developing for around 5 years (about 5 years ago). I wanted to make a powerful objection notation for configuring complex systems and I had an idea to make a more advanced technology demo - to demonstrate the online code editor and code completion. I never started by thinking about text adventures for the demo. I thought

I would model gamebooks and multiple choice games using the language. I can't remember the day when I decided to morph it into a parser based system but I do remember getting excited about it when I made the switch. I still have an early version of the design where locations were described as pages, and each page had links inside to other pages, and so on. Gamebook mode was added and hidden somewhere in later versions Adventuron for those brave enough to try it out, but at the beginning, I started to become more interested in making my own database-based text adventure engine.

I started to model a database-driven text adventure engine, with a side by side view in a web page, all running client-side, no need for expensive servers. It was a database based text adventure engine that tried to emulate the feeling you got from playing a QUILL game (at first). There was nothing too special about the earlier version in terms of functionality. It had a basic parser, but it was easy to code in the browser, and very easy to develop on. There were no users at all for a long time, and Adventuron was private whilst I iterated towards something minimally useful.

I reached out to the (now late) John Wilson in November 2016, asking him if he might be interested in possibly writing something new in "Adventuron". This is what it looked like then (taken from that email):



John had earlier said:

"I retired (officially) three years ago, so there is not much chance of any more adventures ever being produced, however I did put out a couple of 'free' ones in the period since then. These can still be found online and can provide a bit of harmless fun - if you ever feel that way inclined."

Ultimately, John put out another 16 adventures, mostly implemented in Adventuron. John affectionately (I hope) called me out as "the censor" in some of his games for trying against all odds to keep his new games family friendly. The Adventuron versions of his games tended to be clean, and the ports (to the ZX Spectrum) allowed him a bit more freedom. I really loved him and his unexpected death hit me hard.. I think I might have given up without his early encouragement, friendship, and daily emails.

It looks like a major aim for Adventuron is to reach out to new players and creators, with the very nice Adventuron Classroom website, a few game jams, and you also organize the Text Adventure Literacy Project. Could you tell us a little bit more on that topic?

The background goal to reach younger players was there early on. I wanted to create some tooling that was simple enough for beginners to be able to quickly put together a simple game. The idea was that if I could make Adventuron easy to use, then users would arrive, and it would be an ideal candidate for a first coding language. Many coding languages are really boring at the beginning and focus on x and y co-ordinates, collisions, click a button to play a sound effect, etc.

Back to John. He was the first Adventuron user, and at the beginning he struggled a bit. I set myself a goal to make it easy enough for John to have a good time with. Let's make user number one happy - but via configurability. John had a process of writing where he just sat at the computer and wrote. Without a plan. He just wrote and joined it all together. He loved the QUILL best of all for the process, and I think he had tried a couple of tools but it didn't put him in the zone. He was in his late 60s when I first tried to seduce him. I iterated very often and also delivered his beloved (and my beloved) text colouring. but he also wanted his beloved RUSTIC font too, and you'll see his early Adventuron games have a feel almost identical to his classic games. I could sense he was getting in the zone, and I started to believe it was ready for other people to start to look at.

I released Adventuron publicly on December 31 2017 via a blog post on Medium. After that initial announcement a few users came to check out the system. Gareth Pitchford, Richard Pettrigrew were the next two. Gareth wrote quite a few games in Adventuron and I really enjoyed his games (as did my niece). Richard came along and wrote a nice game about crash landing on an island full of dinosaurs (which is the first Adventuron game to be entered into IFCComp actually). John was churning out games rapidly over this period. Ports of old games and new games too. But I was fairly disappointed that very few people knew about the system and it seemed like it had flopped quite badly. Still - I persisted, and I speculated that I just needed better documentation, something integrated perhaps. A huge amount of work went into the next version of Adventuron.

I released Adventuron Classroom edition in May 2019. In this version of Adventuron I baked a tutorial into the editor. I decided to have 3 main panels - tutorial, code editor, gameplay window. I commissioned original art for the UI by Mark Harrision (Illustrator). I developed the code for the tutorial system myself, via a wrapper atop of AsciiDoc, integrating active media elements into the tutorial - as well as a chapter based system of learning that remembers the readers progress - but also provides instant access to the reference guide. This system can even quiz the player on their knowledge to gain access to next chapters. I thought that if I displayed too much documentation at once, it's overwhelming. If there is too little - it appears trivial. The purpose of the integrated documentation was to stagger the learning - and also to cover concepts like how to copy and paste, how to use code completion, etc. I couldn't assume a child would know any of this information. I'm fairly happy with how it turned out. The tutorial guides students into creating their own 8-bit style illustrated text adventure game, step by step. I used "Excalibur: Sword of Kings" as the basis for the tutorial, and went so far as to buy the licence for the game from Alternative Software, as I was a big fan as a youngster.

Still, even after releasing Adventuron Classroom, the needle barely moved. There was a bit more interest, but not enough to cover the development time (Adventuron itself had almost doubled its feature set at the same time). To really get traction, I would have to build the community (more on that later). I started writing games myself, with my favourite three games being TWO, SPOOKY ADVENTURE and THE PATH. I also ported EXCALIBUR, THE BEAST, HAMURABI (which is a non text adventure game running in a while loop in Adventuron) and a few more. I thought that I could kick start the ecosystem with some content.

The Text Adventure Literacy Project is basically another post on medium and across other media platforms that calls for a common set of design principles that text adventures can adopt when designing games for absolute beginners. TALP games should have an integrated tutorial that walks users through all the common commands, navigation, and how to solve puzzles via textual (or voice) commands. None of this is absolutely novel. It's been done in other games over the years, but I wanted to raise awareness that people under the age of 40 are being excluded from the genre unless we actually teach them how to play from within our games. The TALP proposal also calls for TALP to be used as a suffix on games that are suitable for absolute beginners - so that search engines can find these titles. The initiative was system agnostic of course, and TALP is not a registered trademark. It's just a mindset really - are you writing for yesterday, or are you writing for today?

How have you managed to grow the Adventuron community? What is the current headcount? Are there any lessons for the text adventure / interactive fiction community to take inspiration from?

The Adventuron community has grown since the days of "The Adventuron One" and there are now over 130 people in the discord, and I would imagine several hundred games exist but over 100 have been documented thus far. I recently became aware that 30 games were produced in just one school as part of a course, all these games being non-public. I've heard that Adventuron is used in team building events and coding clubs. It's not possible for me to detect how many people are using Adventuron as I decided to make access to the system free of logins and DRM. I despise active DRM. I'm not against adding an optional login to Adventuron in the future but I'm not going to support active DRM in the engine itself. I also uploaded the most recent (non beta) version of Adventuron to archive dot org, which runs entirely offline. I know some people have reservations that I'm going to turn evil and switch it access to the editor, but it's there on archive dot org along with all documentation and samples. As a rule of thumb, it's good to be suspicious of online systems.

Getting back to community - the way I grew the community was to hold regular gamejams. These are competitions where game authors are challenged to write a game within a particular set of constraints. The first gamejam was called "The Adventuron Cavejam". The compelling aspect of this jam (I think) was that I provided a skeleton of a game, with the jam being the challenge to remix it and turn it into whatever kind of game you wanted (that was also child friendly). Since then I've

organised six other game-jams, and each jam takes a lot of effort to organise. The premise, the prizes, the prize fulfillments, the fixing of bug reports, helping with questions to do with design, etc. But I enjoy it, and recently, the community is supporting itself - which is so wonderful imho.

Oh, big shout out to Mark Hardisty for donating physical copies of "The Classic Adventurer" bookazine to every gamejam and Ricardo Oyón for creating much of the banner art associated with many of the jams. Also big shout out to ifMUD community for a massive cash injection into the recent gamejam (more on that later).

I suppose the lessons I would give for anyone wanting to grow their IF/TA (system) community is to find a USP, to provide good documentation / tutorials, to create a community hub, to provide access to yourself, to appreciate the community, to fix bugs promptly, to empathise with your users, and to focus the community on regular events. Adventuron is good in a few key areas, and bad in quite a lot of areas, but the game creators seem to enjoy being part of something that feels alive.

You recently added the functionality in Adventuron of being able to export some games (not all Adventuron features are supported) on Spectrum via DAAD. How has the response been? Do you find that your players/authors generally interested in retro text adventures for old computers, or are they content with it being a design aesthetic and not have to worry about available space or limited functionality?

I think the function of exporting to 8-bit platforms is really interesting from a technology point of view. How could Adventuron, which itself is a 2000K blob of JavaScript, be used as a tool to export games that have to fit in 48K of RAM, including the engine and screen RAM itself? Of course the answer was to transform the data to existing engines, and DAAD was a perfect choice as it had been recently put into the public domain by Andrés Samudio, DAAD previously being a private commercial product. The problem is that Adventuron is a superset of DAAD, particularly with or expressions, if-then-else statements, barriers, and that kind of work. In some ways DAAD too has other features that Adventuron does not have, so there is a mapping operation that needs to go on between Adventuron and DAAD.

"8-bit mode" limits Adventuron game source code to the subset of features that can be mapped to DAAD. This mode tries to optimise system message mappings, and also automatically converts images to the format required of DAAD Ready such that

games can export to ZX Spectrum, ZX Spectrum Next, DOS and more. DAAD is a powerful and efficient product, but it's quite complex for the beginner, being a very raw syntax. Adventuron just really wraps up that complexity, and makes it easier for beginners to leverage it (along with the excellent Maluva library by Uto - which adds support for disk or sd based messages and graphics).

I personally think that games that target 8-bit computers are not the future of the genre (in 2022). It's a niche of a niche of a niche. There is certainly an audience there, and it was a fun technical problem to solve, and I love seeing games that result from this path (*Golden Seas* for ZX Next is amazing) but I personally think the future of the text adventure genre is something completely different, or at least, significantly evolved, without losing the core features of the genre. That isn't to say that 8-bit mode won't be getting its own revolution over time.

How do you see Adventuron evolving? What do you put your energy towards, these days? How can my readers get involved?

Adventuron is evolving rather slowly now, but it is evolving. There are two strands really. One is a push towards the modern, especially in terms of the UI and UX. The other is a push towards a refined version of traditional engines. For the modern path, there are many innovations that are currently underway. Auto mapping, speech controlled UX, modern typography as standard, voxel support, zero keyboard gameplay, gamebook mode, more modern renderer / pixel shaders / animation system text flow system. Additionally Adventuron needs to become more powerful as an engine, with better support for containers, better scoping, improved syntax and more.

The other direction is an evolution of 8-bit mode. I'll announce more of that later this year.

There is a new gamejam on at the moment, "The Text Adventure Literacy Jam (2022)", with lots of prizes. I think a great way to get involved is to build adventure games, in any engine - not just adventuron, that can appeal to younger players. At time of writing 60 authors have signed up. This generally translates to about 15 games that actually deliver a game (on Itch). I think that's a really nice crop, and there are a number of different systems represented. I'm happy to see what games and innovations lie in store.

Growing the audience for text adventure (or IF games) is not a zero sum game.

In an unsaturated market, we all succeed when we help everyone else succeed.

I'm excited to see where Adventuron and IF/TA goes in the next 5 years.

Thanks for the interview.