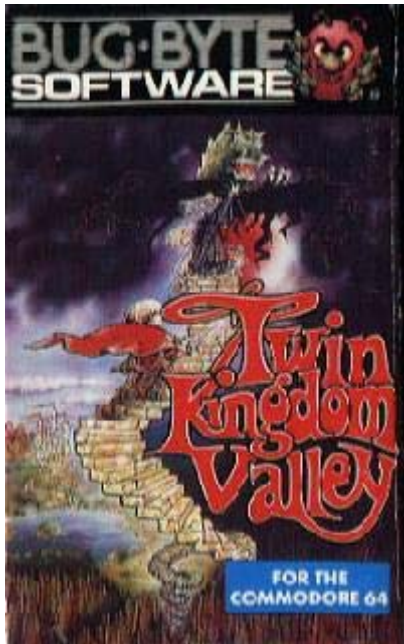# Trevor Hall, Twin Kingdom Valley and the rise of the NPCs

1983 saw the introduction of a new era in home computer adventure gaming: NPCs [non-player characters]. (Sure, the Zork titles beat the competition by at least a year, but they cost an arm and a leg and would only run off a ridiculously expensive disk drive - yes, people, those were the days!)
The Hobbit is probably the best-known of the pioneers, but Trevor Hall's Twin Kingdom Valley certainly made a mark, too, sporting dozens of characters roaming the hitherto unpopulated landscapes and bringing life to an otherwise often very static genre. I've thrown some questions at Trevor to hear what life in the pioneering business was like back then.

Tell us a bit about your background

I was born near York, 1959. I studied computer science at Manchester University. There I got hooked on the arcade games at the students' union and got an "Acorn Atom", (Amazing! A whole computer just for me!) and started to program (in 6502).

How did you get involved with writing adventure games in the first place?

The early game [Crowther and Woods' "Adventure"] was legendary, but only on "big computers". It seemed easy to describe so some friends and I sketched some maps. I then postulated how trivial it would be to describe a location on the map, and connections to other places "in a few bytes". From there, a text location list was made.
Next: the parser: having studied compilers, and played other games, this was the easy part. Just tokenise the text, and recognise a verb and an object.
Next: the graphics. I really wanted "real 3D", but could not figure that out, much too slow. I studied line and circle drawing as course work. Flood fill recursion didn't take long to code.
I compromised on drawing scaled pictures to simulate 3D. The pictures were carefully designed to reduce overlap of objects, because I was using flood fill instead of solid triangles (as in current 3D cards). A friend drew many of the objects (and got paid when the game was sold, a fixed fee). All of the assembly code was easy to do, having done "Space Invaders" before.

TKV was one of the first games to feature NPCs. Did it come out before or after The Hobbit (I really can't remember - they were both released in '83)?

The game was developed about the same time, but I think The Hobbit came out first (on Spectrum). TKV came out soon after on BBC, then got ported to C64, then Spectrum.

So, were the NPCs mostly introduced as a programming exercise (a quite impressive one, at that), or did you see it as an integrated part of your game?

It was an essential game feature. Implementing it was an interesting exercise.

How did you handle the NPCs? In addition, NPCs - at least in early titles - often seem to have a gimmicky quality to them. Sure, they roam the countryside and give you a sense of activity in the game, but they really don't do anything *for* the game, except waste the player's time since he has to run around looking for them all the time! How did you go about this in your game?

In other games, the characters were lame. They appeared to come and go. You could not give chase. I decided the game should have more active players.

The game actually had the following "tables" or "databases" with interpreters for each (and assemblers to code the game):

Location database: up to 256 locations possible, each with a description, and list of exits (a direction code byte [including locks]+ 1 byte target location). One byte "image number" per location. Locations also have a zone.

Object database: object with weight, value, throw damage, hit damage, fragility and location.

Creature database: Creatures with location and inventory of objects, friend/foe flag, permitted zones and health. One of the creatures is "You".

Graphics database: 256 objects, (including sub-objects), drawn on a 16x16 grid. Giving about 80-100 location images, using about 200 objects in the BBC version, almost all 256 objects used in expanded C64 version.

So: Your question on NPCs.

They could inspect the tables (except pictures, no need for that).

They have a decision tree including some random numbers, but basically:

Can I pick up a weapon?

Can I fight?

Should I flee?

Should I wander (in my zone)?

All their fight chances were similar to yours, except "You" always got the first blow.

Note that even if you chose to run away, the other blows still came.

The game had a combat exception rule that running if your last action was to fight, let you run away.

All in all, the rules were probably less than 1/2 K of code!

I particularly liked to rescue and arm all the little guys in the castle dungeon (one of my fave bits of the game). It was kinda nice to free all those guys and get them to help you fight the good fight.

What did you use as inspiration for TKV?

The original adventure

TKV is one of the finest early collaborations between role playing games and traditional IF. Have you ever played any RPG games?

Not much. My friends at the time played "Dungeons and Dragons" (D&D, I think similar to AD&D) and roped me into a few sessions, accompanied by large jugs of cider etc. This helped. I think I like my actions to be mostly natural (like "throw") instead of opening a door by casting a spell. Many RPGs are a somewhat tedious repeat of the player level and spell casting stuff from D&D. You spend a lot of time figuring our what the spells and powers are. Considering that there are 1000s of things which a person can already do to simulate, I prefer to expand vocabulary of the game until English is exhausted, before going too mythical.

The end of the game ends, rather mysteriously, with the player discovering the Secret of Life. What *is* the Secret of Life? [wouldn't we all like to know?]

The Secret is told to you after you type "view secret" or "look secret".
First the image (recursive graphic) is plotted. Then the game tells you "The secret of life is Stay Cool", and the game (if I remember correctly) ends.

The early computer scene was filled with small companies whose marketing strategies were limited to crudely drawn ads (I'm not referring to the very neat TKV cover) for simple BASIC shoot'em-ups - a far cry from today's multi-million dollar games. What was it like being part of an emerging gaming industry? How did you manage to get heard, with so many companies releasing games of so varying quality?

I think I did what many others wanting to publish did (and presumably still do):
1: Find a game publisher who already exists, and has some titles available, and adverts in mags etc.
2: Send the game for evaluation.
I think Bug-Byte was the first I tried, and they offered a good royalty rate. I don't know where the company got its start up cash, but there were 3 main players who handled (or mishandled) all that. They had some advantages, like a good artist, a very good deal on rent in a new unoccupied building, their own tape duplication machines. They built a good name from the Spectrum Manic miner series, BBC micro stuff from me and others, and some C64, but lost the Manic Miner programmer to another company. That was probably a bad turning point for them.

I tried looking up your name in the Giant List of Classic Game Programmers, and your name only came up with this one entry? Does that mean that TKV is your only game?

You should look up the Bug-Byte archives for BBC micro. All the other games by me (sole author) are "shoot-em-up". They include:
Oblivion, Space Invaders (yet another clone!), Galaxy Wars, Sea Lord, City Defence.
They're not that hot, but I liked Oblivion the best.
For the company:
Automan (based on the TV series). Note: not found this on downloads yet.
Star Trader (interplanetary trading game with some combat)
There was once going to be a sequel to TKV called "Valley of Gold", but I gave up when Bug-Byte collapsed owing me oodles of royalty.
Maybe someday I'll do the sequel. It would work well on the new Nvidia chip.

You told me that your involvement with Bug-Byte didn't go as smoothly as could be hoped for. Something about missing royalties?

They went belly-up, owing me and many other creditors.

Did you ever earn anything from writing the game?

Yes, for a new graduate in Manchester in his early twenties it seemed a princely sum. Probably the last 3-6 months of due royalty were lost as the company went under.

What have you been doing since the BBC days?

Mostly mass spectrometry (analytical chemistry). In other words scientific software for a specialist industry. (For more details on the subject try thermo.com)
My current job title "Software manager, Finnigan, San Jose office". Not surprisingly, the job includes a fair amount of graphics.

What do you think of the game scene today (as compared to the one in the early 80s)?

I think "Homeworld" has slightly better graphics (cough, cough...). [and it's quite fun, too, not exactly an adventure game, though! -JG]
If I had the 3D cards then, I'm sure I would have done better.
Still, many games have lots of hi-tech, and not that much gameplay. I like the mystery in the old adventures.
I think the Tex Murphy series was good (though it seems to have dried up). For example "Killing Moon" was a good pace. Just like an adventure, you could explore, pick things up, and not get attacked every three seconds by bats and goblins.
The original "8-bit" games were written for the fun of it, and sold for royalties. Now the commercial aspect has taken over so many of the titles. The worst of the lot are most (but not all) movie games. I quite liked Blade Runner, though.
With the advent of freeware, perhaps some more of the imaginative games "written for the hell of it" will appear?
Well, check out the IF-archive's TADS and Inform sections. Lots of imaginative titles there!

Thanks to Trevor for his insights.
Jacob Gunness, November 1999