Adventure Workshop

AMOS COLUMN

STAC: The simple solution

HEAD'S CASES: SURE FIRE FUN

ISSUE 1
£1.25

You're reading the only magazine dedicated to adventure writing on 16-bit computers! So if you use an Amiga, an Atari ST, or a PC then you've come to the right place. Not only will you find specific articles for your machine here, but also a range of items on more general aspects of designing, programming, testing and even selling your adventure games! But this isn't an elite magazine staffed by experts only - it's YOUR magazine! Whatever you have to offer, be it a small tip you've found useful, or a point you wish to debate, your contribution WILL be welcome! All the contents of this first issue are from readers who have chosen to share their knowledge. If we don't help each other, then we leave ourselves isolated and vulnerable to problems. You might have written a good routine yourself that someone else is struggling without! What's easy for you may be hard for others. By pooling our knowledge, we can assist ourselves, with the result that better adventures are written easier and quicker. Whatever you want out of Adventure Workshop, you're bound to find it here. Forget the large 'glossy' magazines - when do they ever cater for your needs? They prefer to cover arcade games and consoles. They're led by the latest market trends. 'Adventure' is a dirty word to them, and 'Adventure writing help' is almost unheard of! This magazine isn't led by any such chains. It's led by YOU! For YOU, by YOU, with myself merely as the overseer of all your great work. I couldn't do an issue without you - the reader. Yet I am always on the lookout for more contributions. Go on, send something in! Even if it's only a letter.

ADVENTURE WORKSHOP Issue 1 (C) Christopher Hester (EDITOR) 1991. ADDRESS: 3 West Lane, Baildon, Near Shipley, West Yorkshire, BD17 5HD, UK. COVER DESIGN: Editor. SUBSCRIPTIONS: £1.25 Britain, £1.75 Europe + Elsewhere (Seamail), £2.25 Elsewhere (Airmail). PAYMENT: Cheques/POs to "C.HESTER" only, or UK stamps for small amounts. BACK ISSUES: Not yet! ADVERTIZING: £6 Back Cover, £5 Inside Full Page, £2.50 Half Page, £1.25 Quarter Page. SUBMISSIONS: Preferably typed on white A4 with 15mm border. Use dark ribbon + best print style. COPYRIGHT: Remains with named author. Unnamed items (C) C.Hester. FORMAT: Amateur bimonthly, non-profit magazine. COMMENTS: Contributers' views may not be shared by the Editor. DESIGN: Editor. INHOUSE SOFTWARE: Mini Office II (Database). PHOTOCOPYING: John Rodrigues. BINDING: Mandy Rodrigues.

Version 1.0

The Tales of Talespin

by Brian Pell

I've tried my hand at writing adventures on the old and faithful Speccy. None ever hit the world by storm. Then I upgraded to the Amiga and with my love of the games and art I wanted to put the both together. I looked around for a 16-bit version of the PAW and Talespin caught my eye. From the pre on the street hype it seemed the program for me. So with the hammer in my hand I commited first degree murder on my piggy bank and bought it. Once the stars of fame and fortune of a top ten seller had disappeared from my eyes, I focused on the monitor and loaded Talespin.

I was greeted by a friendly female voice who wanted me to play with her, (Where can I get a copy of this program?! - Ed) but I put those thoughts out of my head and sat down and looked through menu after menu. Slowly I found that my ideas of adventuring wouldn't fit into Talespin, I had to change adventures as we know it. I was boldly going where no man had gòne before. The great graphics I was drawing on that fab program Dpaint were knocked for six or 16 the amount of colours you could have on the screen at one time, and the simple commands the package had to offer. But with a different approach and some preplanning my art soon turned out quite good.

Next was the game plan as said Talespin made me rethink. Objects could be picked up yes, but could only be picked up and dropped where you wanted them to be and not anywhere like text adventures. You see objects are small pictures or bubble text which are put in a location. Talespin works on pages like a book and not locations so each page has to be drawn and to drop an object anywhere would mean to put a picture in every location just incase you dropped it there. The picture would only appear when you wanted it to but the memory would be too great. Also, the how many objects one could carry (ie: an inventory) is impossible. Can an adventure be written and played you ask, yes. I overcome these problems by first making you pick an object up and drop it only where you found it and or if you pick an object up one would be dropped in its place.

With the game mouse-driven players didn't have to use their grey matter to find the right words to type in to ie: fix the bulb into the torch. You have to give a choice of words to pick from. The game now became a Steve Jackson pick a page adventure game. What gives the game a big difference is that you can touch objects to look, examine and even take them. Drawers can be opened and looked into, objects can be hidden behind things. To save memory each location or page has its own value, with this set at different numbers you can put a number of pictures in it which only appear when you want them to, this means each page can be several locations.

One of the new ideas is you can open a door and see it open and pass through if you have the right key, no more endless typing UNLOCK DOOR WITH KEY.

My first game using Talespin is called "The Turdman", a gangster adventure for a 1Mb Amiga and comes on two disks and for a small sum of £9.99 you can play it.

A WHOLE STAC OF PROBLEMS

by Matthew Conway

Part The First: Special Condition 17 And All That Jazz

Before sitting down and typing this article, I thought to myself, "What shall I do to start with?" Of course, it didn't take me long to work out the answer - it's quite obvious really: I shall start at the start!

The start in any STAC adventure is determined by what code is put into Special Condition 17. However, this can't just be any old code because S17 (as I shall call it to save valuable ink and sanity) determines the very first things that the player will see of the actual adventure. A loading screen is all well and good, but the most it can do is to leave the player with a sense of awe at the artist's graphical prowess. Such a feeling, however, is grossly detrimental if the next thing to come up is an awfully-designed introductory screen which appears to have been given all of ten seconds' thought and a similar amount of time in actual programming. As anyone who has ever reviewed an adventure will know, this is as bad a start as can be imagined and ratings are invariably affected, irrespective of the adventure's true virtues. First impressions are vital, and this article is designed to try and help you get the most out of S17.

Before getting down to the code itself, it is vital that you make a list of everything that you are going to put into S17. This may seem quite unnecessary because, after all, if you're clever enough to write an adventure then surely you must be able to remember what you are going to put into that adventure, right? Wrong! Try to do this and you will invariably find that the little ideas which make a good game a great game begin to slip your mind, and there's nothing worse than knowing that you had a great routine just begging to be used and then forgetting what it was. Whenever you get a good idea, scribble it down on a scrap of paper so that you won't forget it. Believe me, having a permanent record of that routine will soon prove to be a blessing and not the waste of time which it initially seems to be.

Anyway, back to S17. This condition must contain a) anything which is set up at the start of the adventure and b) anything to be printed on the screen before the adventure proper begins.

To elaborate, a) includes the initial colour scheme, the screen mode depending on whether the game is text-only or incorporates graphics, the cursor size, counters initially set to values other than 0, markers initially set and not reset, the player's strength value etc, while b) includes the title screen, request for instructions and background etc.

Okay, so you've made a list of what to incorporate into S17. What next? Well, now comes the time for turning all that English into something STAC can understand. Don't worry, though, because I'm not going to advise that you do this on paper first and then transfer it all to the computer - this is one time when putting everything straight into STAC works. However, do take care that you do things one at a time. Never move onto the next item on your list until you have fiddled and tweaked with the previous one to your total satisfaction. Another thing to do is always place separate ideas on different lines. There is no advantage in lumping the entire condition onto one line and then realizing that you can't work out where one item ends and the next one starts - lay things out logically and clearly and you will never have any problems. If need be, take advantage of the fact that you can place comments on these lines to label them

- there is absolutely no excuse for forgetting the purpose of a line of code if you do this!

As an example of what I think a well laid out and effective S17 looks like, here is one of the sort that usually adorns the start of any adventure I might end up writing, along with the comments of what each part of the code does:

```
setstr 100
0 colour 777 3 colour 0 0 topcol 777 3
topcol 0
text
split
message 1 lf lf message 2 lf lf message 3 lf
lf message 4 lf lf lf lf lf yesno then
special 19
```

What this does is this:

setstr 100 - set the player's strength value to 100 units, a convenient number for expressing objects' weights as a percentage of what the player can carry.

0 colour 777 etc - set the initial colour scheme to an all-white background with black text, the easiest to read even if it is somewhat uninspiring.

text - I rarely work with graphics and so consequently the screen mode is set to text-only.

split - switch to 80-column mode, perfectly usable if a legible font is created.

message 1 etc - print the name of the adventure, where the idea came from, who wrote it, the copyright notice, and ask the player if he/she would like to read the background to the adventure.

if yesno then special 19 - if the player wants to read the background, jump to Special Condition 19 which allows just that.

Of course, there are very many other things which could be done in S17. As you begin to implement more and more complex routines, new initial values have to be set. For example, if you wish to implement a routine to force STAC to print "and" before the last object in a room location (a routine I shall be writing about in the near future!), a series of 'mess$' commands have to be placed in S17. Other things crop up all the time - you notice that you need to preset the value of a counter, so that goes in S17. A marker needs to be set before the game starts - S17 again. The range of usages for S17 is enormous, so make sure you make the best of it.

Before this article comes to an end, a few words about what makes a good adventure a great adventure are probably in order, connected as they are with the whole idea of starting things off.

Most importantly, though I hate saying it and purists will probably die of sudden seizures, is the inclusion of graphics. If you have any hopes of getting an adventure picked up by a top software house, graphics are 99% essential. Unfortunately, only a select group of aged adventurers who have been around since the days of "Colossal Adventure" find text adventures to be the best sort there is - the modern adventurer, demanding his machine to be stretched to the limits and his/her eyes to be assailed by all sorts of graphical delights, believes quite the opposite. Which all goes to indicate that I'll never get rich through this programming lark.

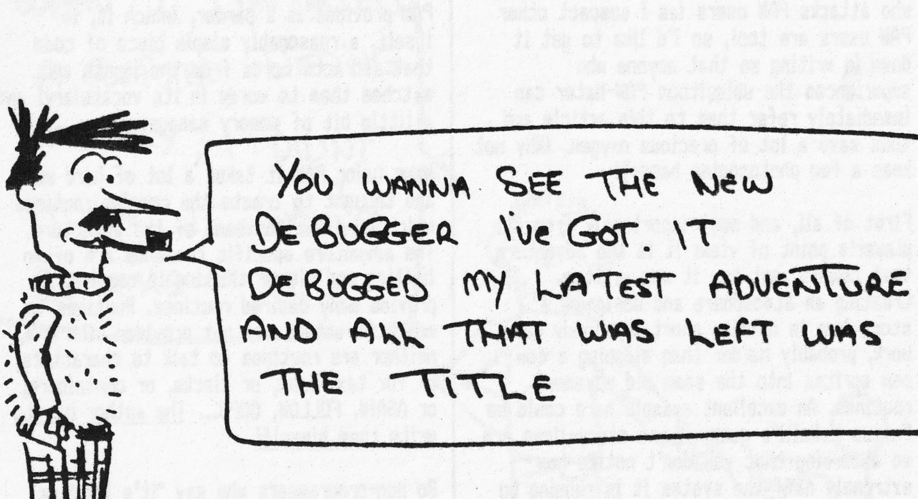
Anyway, personal problems aside, a decent plot comes next. Originality is not as important as some would have you believe, but a credible background which excites the player's imagination is. If you can carry this on into an adventure, as much fun can be gained by exploring the game world as from trying to solve the problems which abound in it.

Thirdly, do not, under any circumstances, include sudden death routines. Never, never, never. No way, no how. Sudden death in an adventure means sudden death for an adventure. There is nothing more infuriating than solving a really vicious problem only to be killed because you forgot about the large monster to the north which you weren't told about and which gives you no chance of escape. Avoid.

Finally, spare no expense at making the parser as responsive as possible. "You can't" helps no-one. If there is some reason given as to why the player can't do what he/she just typed in then he/she is more likely to play on than just sit back and give up. Nobody expects a STACed game to respond like an Infocom adventure, but the nearer you get to that level then the nearer you are to producing a very good game indeed.

Well, that just about wraps up this little discourse on how to get the most from the start of an adventure. If anyone has any comments they would like to make about what I have said here, or if anyone would like to send me any routines which they think would benefit the rest of the readership, or if anyone has a query about an aspect of STAC with which they are not totally at home, then just drop me a line. However, if you would like a personal reply outside of these pages, please send an SSAE as my postal costs are high enough as it is!

Matthew Conway, 1 St.George's Terrace,
Station Road, Lambourn, Berks, RG16 7PW.


YOU WANNA SEE THE NEW DEBUGGER I'VE GOT. IT DEBUGGED MY LATEST ADVENTURE AND ALL THAT WAS LEFT WAS THE TITLE

Programmer, Or Just A Writer?

by Ian Eveleigh

I don't know if you've experienced this, but as an adventure writer who uses PAW I am becoming more than a little miffed off with the way that people who write adventures using PAW, GAC etc, are continually ridiculed by everyone and anyone. Utility written games are down-graded by many reviewers and are even dismissed as totally unacceptable by some software houses (you know who I mean!).

Having spent over five months slaving away over a melting computer writing "Sherlock Holmes - The Lamberley Mystery" (there, I got the plug in!) I get more than a little annoyed when people take the attitude that it was five months of sitting back and doing very little when they discover I used PAW to write the bulk of the adventure.

I am becoming tired of presenting the following argument to each and every person who attacks PAW users (as I suspect other PAW users are too), so I'd like to get it down in writing so that anyone who experiences the ubiquitous PAW-hater can immediately refer them to this article and thus save a lot of precious oxygen. (Why not keep a few photocopies handy?)

First of all, and most importantly from the player's point of view: it is the adventure that counts, not how it was written. Creating an atmosphere and designing a story-line is nothing short of bloody hard work, probably harder than slapping a few new sprites into the same old movement routines. An excellent example here could be Fergus McNeill's games - the story-lines are so absorbing that you don't notice how extremely naff the system it is running on is.

The ignorant consensus seems to be that PAW actually writes the adventure for you; this is absolutely not the case. There IS a large amount of programming skill required to write a quality PAWed adventure. As these critics have probably never used PAW I shall give a brief description of what it is. PAW is, in essence, an adventure writing language, not a utility. It consists of a set of 108 commands specific to writing adventures (and no other kind of program) along with 500-odd messages (which could be considered the equivalent of BASIC strings), 250 location descriptions (also like strings) and 250 flags. The only thing that PAW provides is a parser, (which is, in itself, a reasonably simple piece of code that extracts words from the inputs and matches them to words in its vocabulary) and a little bit of memory management.

> *"it is the adventure that counts, not how it was written "*

When using PAW it takes a lot of hard work and thought to create the complex routines which form the backbone of the adventure. The adventure specific commands are often limiting and clever thinking is required to provide many desired routines. Routines to move characters are not provided with PAW, neither are routines to talk to characters, or for taxi-cabs, or clocks, or containers, or AGAIN, FOLLOW, OOPS... The author has to write them himself!

So non-programmers who say "it's lazy" or "you cheat" should perhaps try writing a successful adventure themselves if they think it's so easy. I can guarantee 99% of them wouldn't be able to create the atmosphere or program the routines to make the game work - yes, that's right, PROGRAM - you do need to!!!

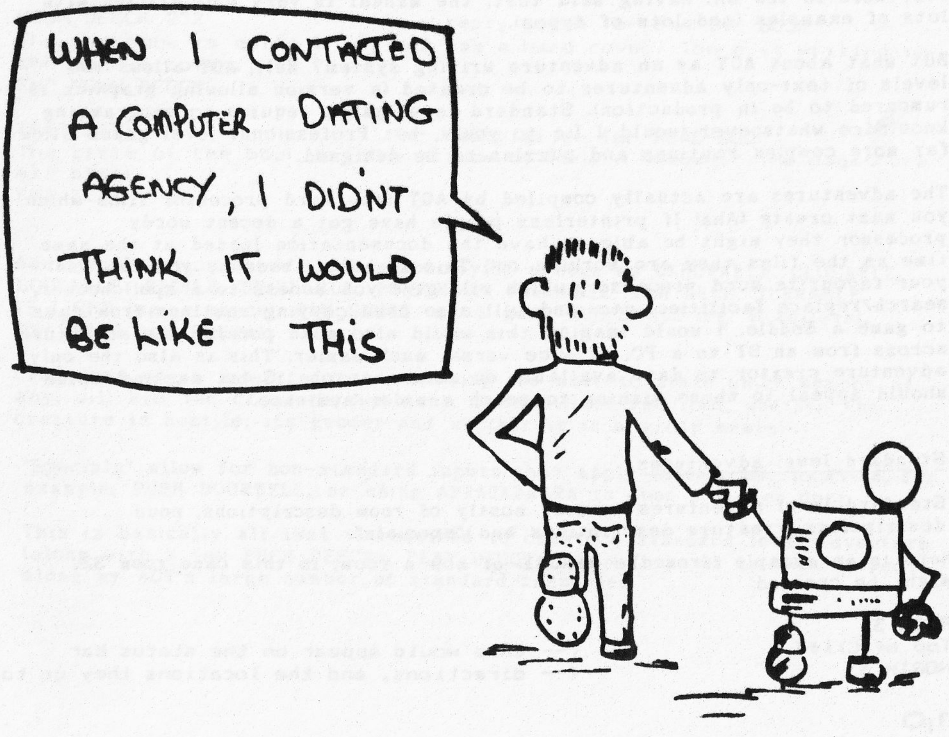
Programmers saying "it's lazy" or "you cheat" should ask themselves why they bother with assembly language when the job can be done easier. With faster and faster machinesbeing developed, assembly language will soon become mostly obsolete, used only to write the interpreters or compilers for other, easier to use languages (PAW could be seen as an early example of this.)

Anyway, nearly all adventures are written using an adventure language of some kind, either a commercial one, or one developed in-house by themselves or a separate programmer (eg: Level 9's KAOS, Infocom's Zil, etc). They all have simplified programming systems - why should someone want to spend hours sweating over a complex assembly language routine when they could write the same routine quicker and easier with a little less brain strain.

Expecting someone to be an excellent programmer as well as literarily creative may be the wrong approach. It makes far more sense to specialize and get the best programmer, and the best designer to work together to produce a better product. Just as many arcade game producers bring together a graphics artist and a programmer and a musician, etc, etc.

Finally, I'll put my money where my mouth is and throw down the gauntlet. If any critic of PAW users can take up a copy of PAW, persist in writing an adventure on a stubborn machine (such as an old 48K Spectrum - nothing flash), get it accepted by 4 software houses and published with reviews and sales figures to match my own, I will publicly apologize and totally take back everything I've said in this article.

Go on then. Prove me wrong...


WHEN I CONTACTED A COMPUTER DATING AGENCY, I DIDNT THINK IT WOULD BE LIKE THIS

AGT Review

by Ian Eveleigh

With Gilsoft still not providing the goods in the form of a 16-bit version of PAW (ST and PC versions have been promised "soon" for a good couple of years now) I thought maybe it was time to start looking for an alternative. Of course, there is STAC, but that's not much use to a PC user like me, but what's this... Adventure Game Toolkit for the PC and ST at a mere £4. Worth a look, surely?

Why so cheap? Well, this is actually a piece of Shareware, ie. software that you are allowed to use for a limited period to see if it suits your needs, then if you become a regular user you are supposed to register with Softworks in America for $20, which will entitle you to technical support and news of new versions.

The goods

AGT arrives in the form of a disk. That's it. This disk contains all the necessary AGT files, the documentation, plus eleven sample adventures (including the classic Crowther and Woods "Colossal Cave Adventure" which everyone has heard of, but no one seems to have actually played! That's probably worth £4 in itself).

Oh yes, I did say the documentation. About 120 sides in total, so get plenty of A4 in and a new printer ribbon. (What do you mean "and a printer"?) This is a bit of a problem; In the US, where the software originates from, you can buy the manual for $20, but that's not exactly that easy to do when you're over here in the UK. Having said that, the manual is very well written with lots of examples (and lots of typos).

But what about AGT as an adventure writing system? Well, AGT allows two levels of text-only adventures to be created (a version allowing graphics is rumoured to be in production). Standard level games require no programming knowledge whatsoever (would I lie to you?), but Professional level games allow far more complex routines and puzzles to be designed.

The adventures are actually compiled by AGT from word processor files which you must create (Aha! If printerless people have got a decent wordy processor they might be able to have the documentation loaded at the same time as the files they are working on!) This is great, because you can use your favourite word processor, which will give you access to a spellchecker, search/replace facilities, etc. and will also make copying routines from game to game a doddle. I would imagine this would also make porting the adventure across from an ST to a PC, or vice versa, much easier. This is also the only adventure creator to date available on more than one 16-bit machine, which should appeal to those wishing to reach a wider audience.

Standard level adventures

Standard level adventures consist mostly of room descriptions, noun descriptions, creature descriptions and "specials".

Here is an example (from the manual) of how a room, in this case room 32, might be created:

```
ROOM 32
Top of Cliff                       <-- this would appear on the status bar
NORTH 33                           <-- directions, and the locations they go to
```

```
SOUTH 34
WEST 35
END_ROOM

ROOM_DESCR 32
You are standing near the edge on the top of a tall cliff. To the east
is a sheer drop of several thousand feet. To the north, west and south
are paths that lead down the side of the mountain.
END_ROOM_DESCR

HELP 32
Be careful, don't go too near the edge!
END_HELP_DESCR
```

More options can be added after the directions to specify whether the room is dark, if the game ends on reaching this room, if there is a locked door here, how many points the player should receive for reaching this location and "specials" (see later).

An example of a noun (also from the manual):

```
NOUN 323
Book
Red
There is a small red book here.
WEIGHT 1
SIZE 3
LOCATION 32                    <-- the object's starting location
READABLE
NOUN_SYNONYMS Cover Title
END_NOUN

NOUN_DESCR 232                 <-- reply given for EXAMINE BOOK
The red book is quite thin, and has a hard cover. There is writing on
the book's cover.
END_NOUN_DESCR

TEXT 232                       <-- reply given for READ BOOK
The title of the book is "The Wisdom of Ronald Reagan." The pages are
all blank.
END TEXT
```

Additional options here include specifying a noun as UNMOVABLE, CLOSABLE, LOCKABLE, EDIBLE, DRINKABLE, POISONOUS, PUSHABLE, PLAYABLE, TURNABLE, IS_LIGHT (eg. a torch), CAN_SHOOT (eg. a gun), NUM_SHOTS (eg. 6 bullets), WEARABLE and PLURAL.

Creatures use a similar format, allowing the user to state which weapon, if any, will kill the creature, how many shots will be required, whether the creature is hostile, its gender and whether it is a group member.

"Specials" allow for non-standard inputs that apply in only one location, for example, PUSH DOORBELL, or using ABRACADABRA to open a hidden door.

This is basically all that is required to create a standard level adventure (along with a few PUSH_DESCRs, PLAY_DESCRs, etc.) The user is then helped along by AGT's large number of standard responses.

Professional level adventures

As you can imagine, standard level adventures are a little limiting. This is where professional level adventures come in. This puts 250 flags, 25 counters, 25 variables, 25 questions, 250 messages and 400 meta-commands at the users disposal.

The meta-command file you can create (to my mind very reminiscent of PAW's response table) allows you to really go to town. This table is checked after every input by the player, but before AGT's standard responses. Here are a couple of examples of meta-commands:

```
COMMAND CROSS BRIDGE
  AtLocation 23               (* West side of bridge? *)
  GoToRoom 24                 (* East side of bride *)
  PrintMessage 24             (* You walk over the bridge *)
  DoneWithTurn                (* Get player's next input *)
END_COMMAND

COMMAND KISS PRINCESS
  InRoom 305                  (* Princess (creature 305) here? *)
  NOT AtLocation 99           (* not in bridal suite of palace? *)
  PrintMessage 46             (* "Not here!" said the Princess *)
END_COMMAND
```

Meta-commands can become increasingly complex, especially with the use of OR (something that PAW sadly lacked), up to the likes of:

```
COMMAND BREAK LOCK
  InRoom 208                  (* Oak door (noun 208) is here? *)
  NOT InRoom 307              (* Evil Wizard (creature 307) not here? *)
  IsCarrying 223              (* Are you carrying the battle axe? *)
    OR
  Present 246                 (* Large two-handed sword is here? *)
  VariableGT 7 90             (* Player has strength to swing sword? *)
  FlagON 3                    (* Sword has been pulled free from rock? *)
    OR
  IsCarrying 221              (* Are you carrying the iron mace? *)
  VariablGT 7 50              (* Player has enough strength? *)
  SwapLocations 208 209       (* Swap locked door for open doorway *)
  PrintMessage 86             (* "Your blows break the lock..." *)
  ChangePassageway 1 25       (* Open a passage north (1) to room 25 *)
  DoneWithTurn                (* No further process for this turn *)
END_COMMAND
```

The comments you see by the side of the commands can actually be entered in the meta-command file (you can use markers of your own choice) as AGT's compiler will ignore them. This should make debugging AGT games a dream.

Now for the bad news. Remember those lovely helpful standard AGT responses? Well, at the professional level they hinder. They are too standard: You are still stuck with the same old "You are now carrying the...", "You could not go south...", "For some reason you couldn't get the gun to shoot...". This infuriated me as it is a case of stupidity on a severe scale by the authors. Until this point AGT had everything going for it, it could have been the utility to put Gilsoft out of business, but it appears that ALL AGT games will have these cloned messages. Aarrgghh! It looks like a major hack mission will be required to change them (or $50 for the Turbo Pascal 4.0 source code (plus 50-odd quid for a Turbo Pascal 4.0 compiler!)) I personally always prefer to use the first or third person past tense, even if you are happy using the second person present test it is still pretty likely that you would want to customize the messages to some extent.

Whilst we're on the moans, another fault is the way AGT displays objects that are present. It does not use a nice tidy:

```
You can also see:-
  a half-eaten banana
  a Garfield telephone
  a complete collection of Marillion cassettes
```

after the location description, but instead does a:

```
  There is a half-eaten banana here.
  There is a Garfield telephone here.
  There is a complete collection of Marillion cassettes here.
```

which I found very odd, as the inventory displays a (near) perfect:

```
You are carrying the following:
  half-eaten banana
  Garfield telephone
  Marillion cassettes
You are wearing the following:
  trench coat
```

Still, this isn't a complete disaster, as long as the game designer doesn't leave too many objects (say, more than two) to be found in one room then things won't look too bad.

AGT's parser is a bit better than PAW's, allowing pronouns (IT, HIM, HER, THEM, MY, ITS) to be used without any additional programming required, although it does lack an EXCEPT (ie. TAKE ALL EXCEPT...)

The display whilst playing the adventure is very tidy (Infocom style, I'm told) with a neat status bar showing the current location, score and turn at the top and the text scrolling beneath. The user can, of course select the colours.

The Number Manager

As if writing an AGT adventure wasn't easy enough already, there is also another program, the AGT Number Manager, on the disk. This allows you to use labels and leave the computer to worry about the numbers. For example:

```
COMMAND KISS PRINCESS
  InRoom {princess}
  NOT AtLocation {bridal suite}
  PrintMessage {not here!}
END_COMMAND
```

is perfectly acceptable, and makes REMarks virtually obsolete. AGTNUM also lets you create just one word processor file, leaving the computer to split it into the three required for compiling (one for room/noun info, one for messages and one for meta-commands).

It is this particular ability of AGT that puts it head and shoulders (and probably most of the upper torso as well) above the competition.

All in all...

Overlooking its problems, AGT is certainly worth getting (especially at this price). With a bit more investigation, and a few nifty routines, it certainly has the potential to be the starting point for many a masterpiece.

For those who are interested, AGT is available from (cheques made payable to) Amazon Systems, Merlewood, Lodge Hill Road, Farnham, Surrey, GU10 3RD for £4 on the PC (state 3.5" or 5.25" disk) and the ST. Note that buying AGT also entitles you to obtain the book Computer Adventures - The Secret Art, for a vastly reduced price of £5 (as opposed to £9.45).

AGT column!

I will hopefully be starting an AGT column on these very pages sometime in the near future. To do this I'll need your support. So why not check out a copy of AGT (at £4 you can't go wrong) and send your problems, ideas or queries to me, Ian Eveleigh, at 45 Sandcliffe Road, Grantham, Lincolnshire, NG31 8EW, and I'll do my very best to provide a solution.

The Ed's Favourite Records Of 1990

ALBUMS

1 "VIGIL IN A WILDERNESS OF MIRRORS" Fish
2 "WELCOME TO THE SHOW" Barclay James Harvest
3 "REQUIEM FOR THE AMERICAS" Jonathan Elias (+ Jon Anderson)
4 "SERIOUS HITS... LIVE!" Phil Collins
5 "ONE WORLD ONE VOICE" Various Artists
6 "THE WALL LIVE IN BERLIN" Roger Waters
7 "THE CITY" Vangelis
8 "NATURAL HISTORY - THE VERY BEST OF" Talk Talk
9 "VIOLATOR" Depeche Mode
10 "GREATEST HITS... SO FAR" PIL

SINGLES

1 "A GENTLEMAN'S EXCUSE ME" Fish
2 "CHEAP THE BULLET" Barclay James Harvest
3 "SOMETHING HAPPENED ON THE WAY TO HEAVEN" Phil Collins
4 "I WISH IT WOULD RAIN DOWN" Phil Collins
5 "EASTER" Marillion
6 "THE UNINVITED GUEST" Marillion
7 "TATTOOED MILLIONAIRE" Bruce Dickinson
8 "THE POLICY OF TRUTH" Depeche Mode
9 "DON'T ASK ME" PIL
10 "ENJOY THE SILENCE" Depeche Mode

Other gems by: London Beat, The B-52s, EMF, ABWH...

ALBUMS TO LOOK FORWARD TO

"The Shadowplay & The Celtic Illumination" FISH - *Need I say more?*
"Page Of Life" JON & VANGELIS - *If they wish to release it!*
"?" LEVEL 42 - *It was supposed to be out last October!*
"?" ABWH - *The eagerly-awaited 2nd album from ex-Yes members*
"?" GENESIS - *Maybe at the end of the year, if it's ready*
"?" YES - *They went in the studio a few months ago, so here's hoping!*
"Databanks" BANKSTATEMENT - *My name for a 2nd Tony Banks' project album*
"Holidays In Eden" MARILLION - *They've previewed songs live already!*

Moving up to 68000 by Gerald Kellett Part 1

For 'home' machines there is only one 16 bit processor to speak of, the 68000, for 'business' the 8088/8086 (and 'upgrades' such as the 80286, 80386 & i486, and clones) dominate.

There is as yet only one 'commercial' adventure writing utility for 16 bit machines, and that for the ST only, that's STAC. PD now offers two for the ST, the existing ADVSYS, and a new one AGW both text only. The AMIGA also has two PD systems, ADL & AMIGADVENTURE.
It is debatable as to whether 'TALESPIN' is an adventure writing system, but this seems to have been a total flop, only the discount and clearence mail-orders retailers are still advertising it, most at £10.99 for ST and AMIGA versions.

Incentive have kindly offered to let anyone who sends them an SAE have a copy of the source code of STAC (neither the mention in this mag or ST FORMAT say how big, how much postage or whether this is only open to STAC owners). Anyone taken them up on this offer ? This should open-up widespread 'tweaking' of this utility and encourage some to have a go at 68000 code, so although I would no regard myself as particularly proficient at this, all of this months column, and parts of subsequent ones will be on 'moving-up' to 68000 programming, mostly with reference to Z80 code but with the occasional aside on 6502.

The first thing to note about the 68000 is that there are only two types of instructions for shifting data about, the major one replaces LD, PUSH & POP and is quite logically called the MOVE instruction, the other is SWAP. One very important thing to note about the MOVE op is that it MOVEs from source to destination, as opposed to LD destination from source.
The 68000 has in addition to the usual 'flag/system register(s)' and program counter (PC) 16 registers, all 32 bits, split into two sets of eight, the data registers (d0 to d7) and the address registers (a0 to a7), it does not however have as such a stack pointer (SP) register, a7 serves this function. But that makes a7 the SP you may think, yes and no, while it is used by the system to store return adresses and such like, and most assemblers recogonise the label sp as a synonym for a7, the a7 reg can be opperated on in just the same way as any of the other address registers, and any of these can be used as data stack pointers, although everyone seems to use a7 as if it were the only SP register, a hang-over from writers used to single SP processors, that and the possible fear of 'colliding stacks!
Neither does it have an accumulator register, any of the data (and in many cases also the address) registers can be used.

68000 code curiously lacks INC and DEC instructions, possibly because of the pre-inc/post-dec means you don't need them for pointer ops, there are DJNZ equivalents for all d regs, which are the main uses for INC/DEC (there being INC versions of DJNZ (IJNZ ?) as well), so the programmers of the chips internal instructions (called its micro-code) did not see fit to have special ops for this; ADDQ #1,r and SUBQ #1,r which have 'embedded' data were seen as sufficient.
It also unfortunately lacks equivalents of the block ops: LDIR, LDDR, CPIR, CPDR, INIR, INDR, OTIR, OTDR, (not that those last 4 are likely to be used much in adventure programming).
Also sorely missing are conditional CALL and RET analogs, you have to use a conditional BRAnch to jump round or to BSR or RTS instead.

In the following: r is any 8 bit; rr 16 bit register [reg] (32 for 68000), n an 8 bit: nn 16 bit number, & a an address, I? is IX or IY For 68000 .B means the data is of size byte, .W word (2 bytes this is the default) .L long word (4 bytes). d? any d register, a? any a reg, rrrr any reg as 32 bits.

```
 Z80                 68000                    comment
 LD r,n              MOVE.B #n,r
 LD rr,nn            MOVE.W #nn,rr
  -                  MOVE.W #nnnn,rrrr        no 32 bit on Z80
 LD A,(a)            MOVE.B $a,d?             could be an a reg also
 LD rr,(a)           MOVE.W $a,rr
  -                  MOVE.L $a,rrrr           again not 32 bit Z80
 LD (a),A            MOVE.B d?,$a             as per LD A,(a)
 LD (a),rr           MOVE.W rr,$a
  -                  MOVE.L rrrr,$a
 LD r,(rr)           MOVE.B (a?),r
  -                  MOVE.W (a?),rr
  -                  MOVE.L (a?),rrrr
 LD (rr),r           MOVE.B r,(a?)
  -                  MOVE.L rr,(a?)
  -                  MOVE.W rrrr,(a?)
 LD r2,r1            MOVE.B r1,r2
 LD rr2,rr1          MOVE.W rr1,rr2
  -                  MOVE.L rrrr1,rrrr2
 PUSH rr             MOVE.W rr,-(a?)          -() means pre decrement   S N
  -                  MOVE.B r,-(a?)           \ no 8 bit or 32 bit      E O
  -                  MOVE.L rrrr,-(a?)        / versions for Z80        E T
 POP rr              MOVE.W (a?)+,rr          ()+ means post increment    E
  -                  MOVE.B (a?)+,r           \as per push
  -                  MOVE.L (a?)+,rrrr        /                           1
 LD r,(I?+n)         MOVE.B n(a?),r           \ .W;rr & .L;rrrr also
 LD (I?+n),r         MOVE.B r,n(a?)           /
 push all regs       MOVEM d0-d7/a0-a6,-(a7)  \can be .B, .W or .L with
 pop all regs        MOVEM (a7)+,d0-d7/a0-a6  /any conbination of a & d

 EX DE,HL            SWAP a2,a3               can be any d or a regs
 EXX                 no direct equivalent, but as most ops can be done on
                     both d and a regs SWAP a regs for d could be used

 SET n,r             BSET.B #n,r              n=0-7, .W;n=0-15 & .L;n=0-31 too
 RES n,r             BCLR.B #n,r
 BIT n,r             BTST.B #n,r
  -                  BCHG.B #n,r              inverts bit
 bit_ops n,(rr)      b_ops #n,(a?)            \bit_ops is any of SET, RES, BIT
 bit_ops n,(I?+n1)   b_ops #n,n1(a?)          / b_ops;BSET, BCLR, BCHG & BTST

 CALL nn             JSR   nn                 nnnn also
  -                  BSR.S n                  \ relative Call, .S means 8 bit
  -                  BSR   nnnn               / otherwise 32 bit.
 JP   nn             JMP   nn                 actually nnnn as 32 bit.
 JR   n              BRA.S n                  as BSR
 JP (HL)             JMP   (d?)
  -                  JSR   (d?)               subroutine call too, so no need
                                              to do PUSH ret address.
 RST n              TRAP  #n                  both used for system calls
```

1) Pre decrement, post increment dec/inc by the size of the data.
2) The 68000 has quite a few addressing modes in addition to those that analog those of the Z80. ie loading numbers directly to memory locations without using register addressing: MOVE #n,$a . d register

offsets to indexing register, with number offset too:MOVE r,n(a?,d?). And there are others, I will deal with them and these in more detail another time.

Might I propose the following convention which will allow easier convertion from Z80 and should mean any routines in 68000 code published in this mag will be consistent and can be 'mixed & matched' with a minimum of problems.

Z80 regs		A	A'	BC	DE	HL	IX	IY	BC'	DE'	HL'	SP
68000 regs	arithmetic ops	d0	d7	d1	d2	d3	-	-	d4	d5	d6	-
	address ops	-	-	a1	a2	a3	a4	a5	*	a0	a6	a7

* - use a0 for this too if not used for DE', BC' tends only to be used to pass parameters later swapped to the 'main' set, but if you need to convert from code where both BC' & DE' are used for address functions you can SWAP d4,a0 perform the ops and SWAP d4,a0 again, BUT not if you need to use both simultaneously.
I'm not sure but you don't appear to be able to manipulate the 68000 equivalent of the Z80's F reg, the Status Register (SR).

Following the convention above here are my versions of the three most useful of the Z80 block functions:

```
LDIR    MOVE.B (a3)+,(a2)+        LDDR    MOVE.B (a3),(a2)
        DBNE   d1,LDIR                    SUBQ   #1,a3
        RTS                               SUBQ   #1,a2
                                          DBNE   d1,LDDR
CPIR    CMP.B  (a3)+,d0                   RTS
        BEQ    EXCP
        DBNE   d1,CPIR
EXCP    TST    d1
        RTS
```

Not that is actually worth making these as subroutines, better to use them as Macros or include where they are required.

CPIR will return with zero flag reset only if found (ie do a BNE after BSR CPIR) this is not the same as Z80, if the routine is included explicitly then the BEQ EXCP would jump to 'deal with found' routine and 'deal with not found' would be at EXCP, TST d1 and RTS being omited. ie:

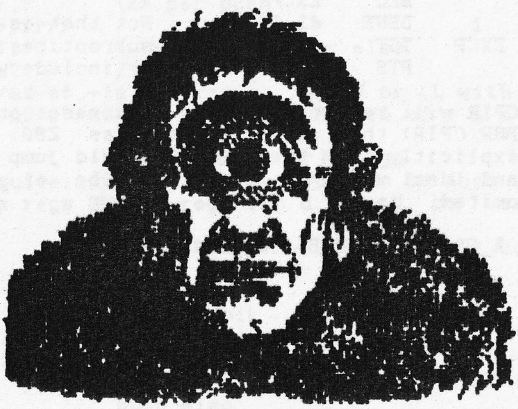
```
A_CPIR     CMP.B  (a3)+,d0
           BEQ    FOUND
           DBNE   d1,A_CPIR
NOT_FOUND  ........
           ........

FOUND      ........
```

S.T.A.C.

The Simple Solution

Part (1)

My friends call me 'One Eye', just a few words about myself...since I wrote a column for Adventure Coder some changes have taken place...my vision started to go haywire so I went to see the Doctor. The Doc said that staring into a V.D.U every evening had finally affected my eyes...double vision...headaches...blurry sight...the doc said that I could expect all these symtoms and more. I was a desperate man, "is there anything that can be done??", I asked. The doc told me about this mate of his who dabbled in eye surgery. This bloke had been struck off FOUR TIMES now if that isn't a good reccomendation then I don't know what is, the eye surgeon said that he would operate for nothing because he had a customer for my eye balls. Without further ado I had the eye of a giant Conger eel transplanted into my forehead. Now I know what you are thinking...where can I get this done??...right?? Well you can sod off, I don't want you lot looking the same as me!!



ONE EYE

That is my 'old look' at the start of the column and my 'new look' is the large pic, it was taken in one of those booths at my local Woolies, not bad eh?? The only draw backs are that I have this terrific craving for Pilchards and I sit with my mouth wide open for ages, but the doc said that it is 'residal nerve response' whatever that is. So it's goodbye to Dave Blower, and hello to 'One Eye'. We are going to make an adventure game using S.T.A.C. It will take many months but will, I hope teach you how to use S.T.A.C. But first some credentials....

I program for a team called Tower Productions, we have one adventure game on sale entitled, 'Souldrinker'. Two other games that are complete but not yet ready for sale are, The Rainbow (Childrens Adventure) and Enday 2240 (Sci-Fi). Tower have been together since October 1988 and consists of, Programmer, Graphics, Music and a Controller who is in charge of text and puzzles. I have never met any of the other members of the Tower team, we communicate via the Postal and Telephone service. so you do not need to live in the same Street or even in the same Town to form a team.

If however you have a vivid imagination and are good at Art and Music then you could do it solo!! What you don't need is any programming skills as I am going to teach you. This column is targeted at the complete novice to S.T.A.C and I will teach you not only how to program but also why it does what it does. What I don't want is to baffle anyone, so if you don't understand any points then please drop me a line, enclosing a stamped addressed envelope. The address to write to is :-

'One Eye'

60 Berkshire Drive

Woolston

Warrington

Cheshire

WA1 4HD

I will require someone to do some graphics for me (Neo or Degas) and also someone to do the music, so if you are interested please contact me.

We already have a scenario for the game thanks to Mat Graiste who submitted his idea via my partially completed column in Adventure Coder. Mats idea was of a 'School' where the central character has been bullied into entering the school after dark in order to get the forthcoming exam results, which will be strewn all around the school in various departments. The idea is simple and I think that is it's strong point. So now we have our scenario....next issue we will start our adventure proper.

Also I may adopt Keith Green's idea as to selling the finished game and donating the proceeds to a Hospital, that is if the finished product is good enough!!

Got to go, my Pilchards on Toast are ready!!

The CES:
An Alternative Report

A (very) personal view of the CES by Ian Eveleigh

Did you get to the PC Show? (Damn! I mean the CES!) Well, I finally struggled down on the Saturday (reducing myself to going down on a plebs day), so here's a summary of what you missed (or rather didn't miss). After one-and-a-half hours driving to London, and 2 hours locating Earls Court (fact: ST owners cannot navigate), we then had to queue another one-hour-forty because I was the only one with a ticket and they wouldn't let me smuggle my two friends in! And me being a totally honest and trustworthy major industry personality! Well, all right, someone who's a bit of a con-artist who knows someone who's got a sister that works in a computer shop that just happened to have an extra complimentary ticket left over! Mind you, you can't beat ponsing around the show wearing a false ID badge and pretending to be incredibly important.

Great failed publicity stunts of the nineties...

Queueing was boring, until Nintendo saw a great opportunity for a publicity stunt. Outside they came: some poor guy dressed up as one of the Super Mario Brothers, the obligatory (American) bimbo, the grey-suited executive and a rather attractive PR lady. They walked up and down the queue for a while, giving out a rain-forest worth's of leaflets. The American then stole a security man's megaphone and tried to start a Mexican wave, or as she called it, a "Mario wave". They started half-way down the queue, working forward... "Go!" Five blokes (just come from a (too) long pub lunch) jumped in the air shouting "Wahaay!!"

"OK. Let's try again," says the bimbo. It passed maybe fifteen people before dying! (This caused a ripple of applause.) The exec has a brain-wave: the people in the queue can't see what the people behind them are doing... so let's start from the front... ten places in front of me! This time they were lucky: nearly 25 people did it! (Excluding myself: I refuse to participate in such cheap publicity stunts (at least not without some form of monetary reward!))

Red faced, the Nintendo crowd go inside: "What we really need is about 15 people to set it off..." says the exec...

The Show...

Inside... well, there was little to rave about. The place seemed smaller than usual and vastly overcrowded with an average number of minute identical stands (with the exception of Ocean, who took up half the floor space (literally)). At points it was impossible to move down the ridiculously narrow aisles between the stands, and people kept tripping over the carpet which collected at the junctions! The show moved to Earls Court in '88 because Olympia proved too small, so I really can't understand why it was all crammed on the (small) first floor, when the (larger) ground floor was unused. It would have made far more sense to spread out on the ground floor. (Keep an eye on the glossies. They usually feature a bird's-eye view of the sparkling show, I bet they don't bother this year).

Ocean, having a monopoly on the software market, also had a monopoly on floor space. Their "stand" featured a small video wall blasting out demos from future games. The "Total Recall" demo consisted of long sections of film (including that great bit where he pulls that bug out through his nose... ugh!) interspersed with (extremely short) game clips (that didn't seem very impressive at all) with "Wicked!" and "Get Down!" plastered all over them.

They also had about 10 arcade machines, a little shop selling the usual games, mugs, posters, Robocop bags, etc., a large and welcome empty space (ah! air!) and an enormous stage with an even larger video wall (which spent most of its time playing Big Bum and Sonia at a nausiatingly loud volume). The stage was for their guest personalities, which included those lovable (?) Teenage Mutating Kung-Fu Tortoises.

Apparently on the Thursday, the guys in the very unconvincing Terrapin suits used these disguises as a method of causing havoc. The Ocean lady introduced them on stage. On comes Leonardo, closely followed by Raphael who proceeds in kicking Leonardo up the backside, which leads to them both jumping off-stage, running round the back of the stand and disappearing into the crowd. (This was <u>not</u> part of the act, and left the woman on stage rather stuck for words). They later showed up on the Mirrorsoft stand (who were celebrating getting the Turtles licence) and started molesting one of their bimbos in a very personal way. She screamed very loudly and ran off into the crowd... Anyway...

Gremlin also had a large stand, mostly dedicated to their new release "Lotus Esprit Turbo Challenge", a rather impressive two player race game. This was available to play, as long as you didn't mind having your driving displayed on a huge monitor over the waiting crowd and being criticized by a sarcastic would-be Murray Walker. A large red Lotus was also on the stand, complete with a "Your Vehicle Has Been Clamped" sticker.

US Gold had a big stand advertising, er... I can't remember! But they had a very nice red over-polished Lamborghini Countach on their stand. I had to be restrained from putting a big sticky handprint on the bonnet.

Whilst on the cars, Mindscape had a Tom Cruise-style race car on their "Days of Thunder" stand, and someone else had a Porsche 911 in full racing livery for some reason or another.

The "new" Amstrad's were there, including the GX4000 console. And those controllers are AWFUL. OK, the machines are nice, but what's the point?! I don't want to tell Alan his job, (but I will anyway) I really think it's time he stopped trying (and failing) to style the industry with his non-standard 3" drives, non-standard Spectrum joystick ports, 25" printer ribbons, not-very compatible not-very fast PC's, and step-back re-styled 8-bit machines based on 1984 technology. It's time he went with the flow. If he doesn't come up with something along the Amiga/ST lines soon (or an impressive, fast, cheap and compatible PC) he's not going to survive in the computer world much longer.

<u>The wonders of Wonderland...</u>

That's about your lot for the big stands, except discretely hidden in a quiet corner of the Virgin stand, attracting very little attention, I found Magnetic Scrolls' Wonderland. This can only be described in one way:

“!”

Now, I'm a bit of a text/graphic worshipper: I hate these fancy new-fangled icon/menu driven adventures. With the downfall of Infocom and Level 9 I thought MS were going the same way by introducing windows. How wrong I was! The PC (VGA) version of Wonderland I saw was spectacular! The mouse was being hogged by a beautiful brunette for most of the day, and I can understand why she was so engrossed! The windows are completely movable, removable and scalable (one each for text, (animated) graphics, inventory, objects here, compass/exits and a map (each location has an individual icon)) and there are menus for verbs, font style/size, etc. But the point is that the game-play, unlike previous menu-driven adventures, has not been "narrowed" by these at all. Hippies can still type-in commands like normal (and even play a standard text-only adventure), but trendies can click items (either from the inventory, etc. or in the picture) then select a verb for it. However, only standard verbs are listed, you still need to type in some of the more unusual verbs, thus keeping the game complex. Magic! Magnetic Scrolls are on to a sure-fire winner here, and will be around for a long time yet!

It was worth the trip just to see this gem in action. However, I'm sorry to say that this really does look like the death of straight text/graphic adventures. Everything will look wet by comparison to this system. Even the hardest text-only fans will not be able to resist the temptations of the self drawing map. (I'll just use the mouse to pick up that object this once...) Oh, thanks a lot MS! You've just killed my favourite part of the software industry!

The rest...

The rest of the show was a triumph in insignificance. About as exciting as a Morrisey twelve-inch (oo-er!). A selection of white box-shaped stands, most of which were dealers flogging cheap software. There were many bargains to be found (3.5" disks at 32p each!!!). But note: only ST, Amiga and the odd PC game were available. Not an 8-bit title in sight!

Adlib gave a worthy demonstration of their (over-priced) PC soundboards. D&H had a popular stand (unfortunately in a very narrow place). The SAM Coupe made an appearance. (Did Miles Gordon go to the Alan Sugar School of Backward Marketing, I ask myself?) The Sega Mega Drive was around, I saw a few people buy one, but didn't see one running (and didn't really want to!) Sierra made a token appearance, with a small stand featuring 2 PC's running recent titles and a table with a few catalogues (but where was Larry 16?). I was especially surprised to see how small Hewson's stand was.

I think I've mentioned just about all the major software houses that were there (mind you, there aren't many left these days).

The other highlight of the day? Whilst sneaking out of the back exit we bumped into Billy Corkhill from Brookside leaving quietly by a side exit and heading towards his Volvo sports car.

The future of CES

All in all, I'm sorry to say the show is gradually going downhill (I wouldn't be surprised to see attendances <u>fall</u> in a couple of years, because regulars get will soon get tired of the bad quality) - it totally lacked the hype and excitement of (say) the '87 do at Olympia. It really needs a good shake to wake it up, and why not move it closer to Christmas, when all the big releases are made, and some bargains can be grabbed for Xmas prezzies?

AMOS Avenue

By John Ferris

Chris has suggested that I have a go at writing an Amiga column for you. As you can see I have decided to give it a go. This article is meant as a basic introduction to AMOS-*The Creator*, the Amiga version of STOS. In later articles I hope to highlight the AMOS commands which will aid the writing of an adventure game.

I bought AMOS a few months ago from Special Reserve. AMOS is very well packaged, the box includes two discs, a 300 page (readable) manual, Issue 0 of the User Club's newsletter and the registration card. I strongly recommend every AMOS user to send in the registration card as Mandarin are taking user support very seriously.

What is AMOS?

AMOS is simply an enhanced version of BASIC, that age-old language which has been written-off by almost everyone who has heard of structured programming. The problem with BASIC was that you couldn't do a lot with it, especially if you had a VIC20 or a C64. The BASICs supplied with these machines were limited to say the least. You were grateful for the ten line magazine listings which peeked, poked and whatevered to simulate the Spectrum's "print at" command.

"Hurray!" cried a hypothetical programmer as he or she saw AmigaBASIC appear on the TV screen for the first time. At last, 16-bit BASIC with all the added features that go with the tag.

"???" goggled the aforementioned programmer as he/she read the program listing needed to load in a picture from Deluxe Paint! All that power but no way for the mere mortal to tap it! Plus bugs! Especially the type which seem to occur at random!

STOS arrived on the ST and set the world alight, "Drool, slobber" went our hypothetical programmer as he/she waited for the Amiga release. "Gasp! wow, goggle! etc." exclaimed the programmer as he/she loaded in King Tut with a *single command* and then scrolled his nose around the screen for good measure!

Get the picture? AMOS is BASIC come of age, a BASIC for serious programming, not just a precursor to machine code. What do you do with such a language? Write scrolly demos and brain dead shoot-em-ups of course!

Ahem, well, er, true, but you can use it for one or two things besides bouncing words and slimy aliens around the screen. You can write..... adventure games! Yes, I've finally got around to the point!

Will AMOS make it easy?

Good question, the answer is no. I would give the same answer to anyone asking the same about GAC, PAW, ADLAN and whatever.
"*I thought* GAC *made things easy?*"
Ask anyone who has written a game with GAC (STAC) and they will tell you that all these utilities do is provide you with a parser, a ready built operating system, a large manual plus lots of counters, markers, flags, conditions....

With all programming languages a little bit of skill is required. "*What level?*" I pretend to hear you cry! Put it like this, hands up all those who have written a game using GAC or PAW. Good, you'll do!

Think I'm kidding? GAC is a series of IF xxxxx THEN xxxxx statements. An adventure game in any form of BASIC is the same, including AMOS.

"*What about the parser and the operating system?*"

Why am I asking myself these questions in italics? In reverse order, the operating system (as I define it) is a loop:

```
10 Gosub initialise
20 If player dead Then Goto you are dead routine
30 Gosub high priority routine
40 Gosub get command routine
50 Gosub parser routine
60 Gosub carry out the commands routine
70 If player wins Then goto lucky *** routine
80 Goto 20
```

That lot is almost the first eight lines of a game I wrote in AmigaBASIC two years ago. It worked as well!

The parser is slightly harder. The hardest part is splitting the sentence into individual words. I cheated, I found a routine in a copy of Commodore User. From that I cobbled together the following code:

```
parser:
 For T = 1 to a: W$(T) = "": Next : V=0:N1=0:N2=0:AD=V
 Line Input A$ : If A$="" Then Goto parser Else A$=A$+" "
 A=1 : B=0
 For C=1 To Len (A$)
 If Mid$(a$,c,1)=" " Then w$(A)= Mid$(A$,B+1,C-1-B):B=C:Inc A :Gosub
 decode
 Next C
```

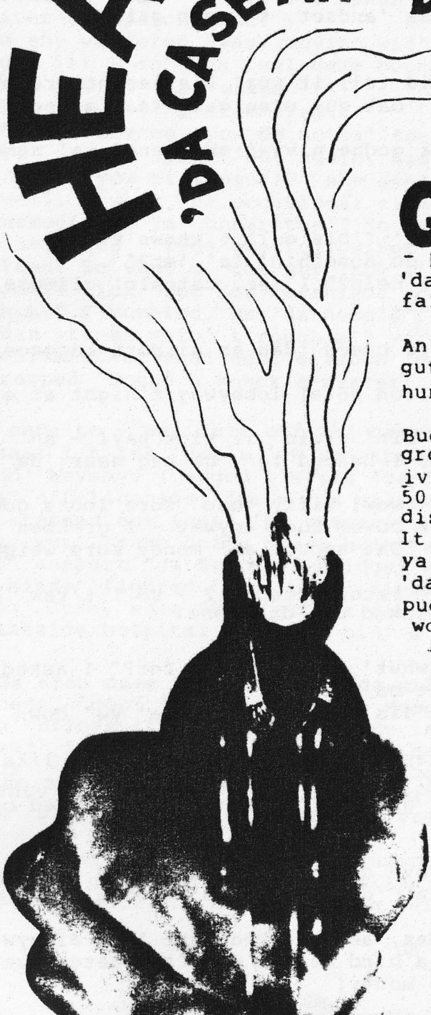
Confusing? I shall explain. The first line clears the word array W$ then sets the verb, noun and adverb numbers to zero. Line two gets the command then checks to see if it contains anything. The program sets two variables. A is the word counter, B is the position of the last space found in A$. The fifth line searches A$ for a space. When found, W$(A) is defined by capturing the characters between the last space (B+1) and the space it has just found (C-1-B). Inc A is a fast version of the command A=A+1. The word w$(A) is then fed to the decoder which attempts to classify it as a verb, noun or adverb. This parser accepts single commands of the type "SHOOT CAT WITH GUN". In other words, one verb, two nouns and an adverb. Similar to the parser in GAC.

That's about it for this article. I agree I haven't said much about AMOS, let alone actually writing a game. The point is that AMOS is very similar to BASIC. To write articles about adventure programming in AMOS is to write about programming in BASIC. I will readily admit to not being a wonderful programmer. Any code I write in this article may not be the most efficient or structured, but it works. If you have a better way of solving a problem then please tell!

In the next article I'll be having a look at the decoder to see just how the W$ array is changed into verbs, nouns and adverbs. I hope to use two AMOS string functions to this end. As an aside, I'll also have a quick look at screens and windows. Have fun!

HEAD'S CASES I: 'DA CASE AH' 'DA NAZI MANGOS

by George March.



Gee' 'dis wuz''da kinda' month I really hated - I wuz' so broke I had 'da only t'ing no amounta' money could buy - poverty! I needed a new case real bad, 'da handles on ma' shoppin' trolley wuz' fallin' ta' bits as well!

An' 'den 'da phone rang.. It said it 'ad gut cramps an' wouldn't be in t'day, I hung up a'fore it could ask fa' sick pay!

Bud' ya' know I shouldn't grouse so much - grousein's fa' 'da Boids! I mean it's not iv'ry day some tasty brunette, bearin' 50,000 dollars free lott'ry money, in a distressed leather briefcase - (distressed? It wuz' nearlly in tears!) - squeezes t'rough ya' door, wid' a twinkle in 'er eyes like 'da tears 'uv an angel, dancin' in 'da puddles on 'da sidewalk, pity t'day ain't wonna' 'dem days either! Bud' she wandered in anyway an' put 'da briefcase down.. Hay! If 'dere's 1 t'ing I love better 'dan a loaded shooter, it's a loaded client!

"Mithtah Williamth?" she lithped, err, lisped, like ah' squeaky castors 'uv a shoppin' trolley on lino'.

"Nah', down ah' hall.." I sez' stickin' a horizontal thumb in 'da air "By 'da way, whut's ya name?" I askes, as subtle as a Jehovah's wittless second 'and car salesman!

"No it isn't!" she squeaked wid' a funny, puzzled look on 'er face "It's Wendy Day!" she sez'.

"Nope, still Friday, right upta'12.00! By 'da way, whut' time d'ya git' off woik?" I sez', real casual like..

"Ha!" she sneers "Don't call me, I'll call you!"

I reckoned she'd 'ave to, I kinda' t'ought 'da phone wuz' still gonna' be off sick fa' a few weeks, bud' even I can be wrong, as 'da phone decided ta' ring again..

I slapped ma' 'and down onna' 'andset - I'd practised 'nough times ta' make it bounce inta' ma' 'and, an' it looked pretty good when it decided ta' woik! Yeah, real good, pity it didn't 'dis time - 'da 1 time I needed ta' show off ta' a real, good broad, I blew it - 'da 'andset bounced off ah' ma' forehead an' ended up inna' trash, 'dat's whut' ya' gets fa' readin' t'ings off 'da back ah' games casettes - as she wiggled outside. Geez' whut' an asshole 'uv a start to a mornin'!

I leant across 'da desk, squintin' inna' 'da basket downna' side 'uv it, fulla' diet soda cans an' 'da screwed up mornin' newsrag.

"Hay!" I screamed at 'da poor defenceless 'andset, t'rough gritted dentures "Whaddaya' want?"

I grabbed 'da phone an' sat down, I could tell it wuz' Lieutenant Drelby onna' line, even a'fore he spoke, geez! 'Dat guy even gave 'da 'andset bad breath!

"Hay, Ritchie Head!" sez' Drelby - t'ank godhe nivva' shortened ma' name!

"Yeah, whaddaya' want, Drelbs?" I asked.

"I got 'dis problem, Head!" 'ey sez'.

"So go see a Doctor!" I told 'im.

"Don't get wiseass wi' me, punk!" 'ey sez' "'Dis gall I knows t'inks somebody's followin' 'er, t'ink ya' can do somethin' fa' 'er?"

"Well I can tap-dance if ya' t'ink it'll help?" I sez, catchin' glimpse ah' 'da dames briefcase wedgin' 'da door open.

"No, you idiot!" he cried.

"Look, Drelby!" I sez' "I've smashed more cases 'dan an airport baggage handler, whaddaya want?"

"Hay!" sez' Drelby "She'll be in 'da Hilton hotel lobbyway t'night at eight, meet 'er!"

'Da phone died - geez' maybe I shoulda' bin' givin' it sick pay! - an' I dropped 'da 'andset back inna' trash, as I headed fa' 'da bag near 'da door..

I opened it up - 'da bag, not 'da door - wow! Fifty thou' sure looks good! An' especially cute in 'dose liddle pink bows! Bud' anyway - I grabbed 'da case an' heaed inna' 'da hallway - hmm? 'Dat amount ah' money sure weighs more 'dan I t'ought it would!

'Da phone went again - funny? I'd nivva' reconnected it? - so I t'rew 'da bag onna' chair, an' sat on it, as I picked up 'da phone.

"Ritchie Head, private h'investigata', whut' can I do ya' for?" I asked, wi' all 'da sublety 'uv a dwarf onna' basket-ball team..

"Yow, Dick!" squeaked 'da voice "I got 'dis real good job fa' ya' down in China town, looks like Chinese maffia!"

"Nah, sorry, Sam, I'm on a real heavy case right now, an' it feels like it's givin' me piles!" I winced, standin' up, I moved it off ah' 'da seat onna' 'da floor, an' replacin' 'da 'andset, I grabbed 'da bag an' staggered out ta' 'da car..

It wuz' night time in 'da city, as I drove downtown, an' grabbed a burger frum' Joe's as I crossed 'da corner upta' 'da Hilton - I'm gonna' 'ave ta' pay 'im back 1 ah' 'dese days!

'Da gun shoved 'tween ma' shoulder blades, as I walked past 'dis alleyway wuz' a dead giveaway 'dat someone wuz' a'hind me - I wasn't a detective fa' nuthin' ya' know! Nah, I did it fa' 'da money!

"Yo, mutha'!" screamed 'da mugger.

Mother? I wasn't even anybody's father, not 'dat I'd own upta' anyway!

"Gimme ya' money, man!" he didn't say much really, I guess 'ey let 'is gun do all 'da talkin' - probably 'ad a better vocabulary! So 'ey grabs 'da case, an' took off down East 33rd and Lexington Avenue, I wuz' lucky I'd taken 'da cash out 'uv it foist!

'Da hotel lobby wuz' full ah' dope pusher's, weirdo's an' ladies ah' 'da night - an' I don't mean nurses! - an' 'dat wuz' only 'da staff!

I caught sight ah' Drelby's dame an' I wanders over, by 'da NY street map tattooed on 'er inner thighs, 'dis dame really knew 'er way around!

"You a friend ah' Donny Drelby?" I asked.

"Hiya han'some, I'm ev'rybody's friend!" she sez', lickin' 'er eyebrows suggestively, an' chewin' 'er gum wid' 'da expoitese 'uv a ruminant wid' a P.H.D. in how ta' look common!

"So, howza' 'bout 50 bucks furra' short time, honey?" she purred.

Hell, I ain't seen 'dat kinda' dough in a hell 'uv a long time!

"Nah!" I sez' "Not t'night, sugar, like, err, cashflow problems ya' know?"

Problems wid' 'da flow? I kinda' reckoned someone musta' built a dam on ma' river ah' paycheques!

So she wandered away, movin' wid' 'da grace 'uv a swan - waddlin' on ice 'dat is! - an' 'da real dame bounced over..

"Ca' savvy?" she asked.

"Urr, whut'?" I stammered.

"Oh! So sorree, yow do notta' speaka' Italiano, no?"

"Ya' wonna' spray 'dat again, kid?"

"So are yow bi-lingual?" she asked, kinda' personal I t'ought, bud..

"Well, I wear 'da occasional sling-back an' stilleto's, bud' 'dat's only fa' chamouflage ya' unnerstan'? So whut's ya' problem?"

"'Zere ees' 'zees mon followingk moy!" she drooled.

"Yeah? So whut's 'dis guy look like?"

"Joust lick 'zee mon beerhinde yow'!"

"Huh?" I mumbled, as 'da hood's fist came down onna' top ah' ma' head faster 'dan prunes goin' t'rough a 2 year old!

Gee! Whutta' real gentle touch 'dis guy 'ad, like a soft, velvet glove, wrapped 'round a knuckle-duster!

I came to - an' fell off ah' campbed - in some scruffy kinda' warehouse - least I t'ink 'dat's 'ow ya' spell it?

Bud' anyway, I edged t'wards 'da door, ya' could tell 'dis place musta' bin' fa' off-licenses, 'dere 'da only t'ing 'dat's open 'dis late at night!

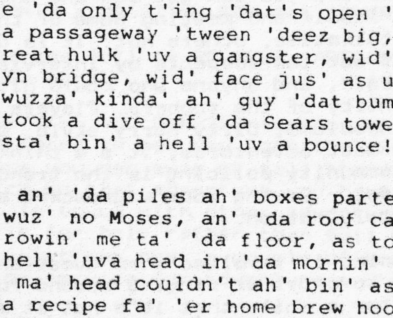
I stepped out inna' 'da passageway 'tween 'deez big, heavy crates, an' 'dere 'ey wuz', some doity great hulk 'uv a gangster, wid' shoulders big 'nough ta' support 'da Brooklyn bridge, wid' face jus' as ugly!

I kinda' figured 'dis wuzza' kinda' ah' guy 'dat bumped off Sam 'da slammer las' June, when Sammy took a dive off 'da Sears tower an' ended up in Eastside hospital - musta' bin' a hell 'uv a bounce!

'Da slob came forward, an' 'da piles ah' boxes parted like 'da sea did fa' Moses, 'cept 'dis guy wuz' no Moses, an' 'da roof came down like 'da ton ah' bricks it wuz', t'rowin' me ta' 'da floor, as tonsils kissed concrete! Geez' I'd sure 'ave a hell 'uva head in 'da mornin' - if I knew where it wuz' 'dat is? Even so, ma' head couldn't ah' bin' as bad as it wuz' after Aunty's las' party, 'da recipe fa' 'er home brew hooch wuz' so old it could ah' bin' wrote by 'da Walton's gran'parents..

An' 'dere wuz' a bit ah' confusion when I h'eventually gotta' heaven, I nearlly got sentta' East Brooklyn instead! - 'Den 'gain, I don't suppose 'dere's much difference 'tween bein' dead an' livin' in Brooklyn!

'Da end..



GAME REVIEW

THE BLAG ST by Tony Woolcock/Geoff Atkinson

ATARI ST/STE price £4.99

Reviewed by Paul Cardin

Originally, 'The Blag' first appeared on QL, apparently receiving many good reviews. Now the authors (two serving policemen) have rewritten and enlarged the game for the ST. It comes on two disks with a huge 800k of program, digitized graphics of real-life characters/situations, and comes with a printed manual, a free pen (which you'll certainly need), a registration form for the competition (more about that later) and a pair of size 14 shoes - no, that last one was a joke.

The player assumes the role of Detective Constable Adrian Blunder of the Beeble Constabulary. You've just arrived to start your first day at sleepy Woolendon police station and find yourself thrown in at the deep end. For a change, you're not hassling the motorist, but getting to grips with a 'blag' or for those not familiar with police/underworld slang, a robbery - from the local bank. This is more like it. Eager to get out and take a few people's particulars down, it doesn't help to find that some clown is hindering your progress right from the start - your car keys have been hidden - and who ever heard of a policeman walking the streets in this day and age? Once you're mobile, you can get out and about easier, exploring the town itself and meeting some of the locals - some friendly and talkative, others not. It is up to you, DC Blunder to sniff out the suspects by interviewing everyone who witnessed the raid, and anyone who could give you some clue as to the identity of the robbers. Players looking to 'blow away' a few hoodlums, Dirty Harry style, will be disappointed. Like most adventures, it's a thinking man's game. Remember - community policing is the trend these days. Be polite and tactful - waving your wagepacket at the local peasants is definitely not on.

A bit of disk-swapping is involved when questioning individuals, going to court or ringing around for information but you'll find after a while that it's not at all intrusive and shouldn't disrupt the flow of the game. In fact, interrogation is a major feature of play. Questioning itself can be a hit or miss affair but if you follow the comprehensive instructions provided, the guidelines are there and you won't go far wrong. It is possible at any point in the game to arrest suspects and take them to court - but you'll need to collect evidence to back up your charges or you'll be laughed out of court and thrown off the case for your troubles.

A good feature of the game is the 'drive' facility. This enables the player to take the car directly to any point on the map - provided it's not somewhere like 'the flat above the garage', you understand. The court-procedure with accompanying graphics is also very stylishly done - you could almost be there.

The game does have niggling errors though, mainly in grammar and text. There are quite a few which could easily have been ironed out to make the game look even more professional. One thing annoyed me about the parser - it didn't understand a lot of the items mentioned in the text - an obvious oversight I thought. A couple of other anomalies occurred - Tina Samon, one of the game's characters, gave her age as 19, yet she looked over 60 in the graphic! Also, at one point, I rang PC Barker at the dog section (geddit?) saying 'Tell me about dogs'. He came back with 'I don't know anything about dogs.'

But, all in all, the good points of the game far outweigh the bad ones and the game has a lot going for it - authenticity, atmosphere and playability being its strong features.

The first player to complete 'The Blag' and successfully convict all the criminals wins £50 worth of software. So, for £4.99 it's well worth buying the game and having a go. And you'll enjoy yourself in the process.

The authors now have the facility to convert the game to one extended format disk for lucky STE/double-sided drive owners for a small charge - no, not loitering with intent - 25p.

Available from: Arresting Software , 60 The Green,
Rowlands Castle , Hants PO9 6AB .

The BLAG ST

Huge Detective Adventure for the Atari ST/STE colour only.Featuring : -

☞ 800k of program.
☞ Digitized Graphics.
☞ Printed Manual.
☞ Software prize for 1st all correct solution.
☞ Supplied on 2 top quality s/s d/d disks.
☞ Reliable extended format used.
☞ Free Pen

You are newly appointed Detective Adrian Blunder somewhat miffed at being posted to sleepy old Woolendon.However all that changes when the local bank is robbed and you find yourself the only one on duty.Are you up to it !There's only one way to find out.

You can question suspects,examine clues for evidence,arrest suspects,track with the dog and when you think you've cracked it,go to court and see what the jury thinks.

Written by two serving police officers you will find it as challenging to play as they did to write.

☞ All this for - ***£4.99*** plus 50p post and packing.Orders despatched by 1st class return of post.

Orders with payment please,to : - G.Atkinson.
60,The Green,Rowlands Castle.Hants.PO9 6AB.

COMPUTER ADVENTURES
- THE SECRET ART -

Reviewed by Christopher Hester

Wouldn't it be good if there was a book on every aspect of adventure writing you could imagine? Well now there is. Author Gil Williamson has come up trumps with an indispensable guide to the field. Whilst you won't find routines listed here, you will find a plethora of useful tips and advice, no matter what your own abilities. Chapter 1 forms an introduction to adventure games, then Chapter 2 deals straight away with presentation and the benefits of either writing a text-only game, or one with graphics. Chapter 3 covers ideas for adventures, themes you could use, even how to base your game on an existing work, such as a novel. The fourth Chapter deals extensively with the plot of your game, and seems to leave nothing out, from invisibility to geography (and mazes of course!). There's simply too many things to list here. You'll be surprised adventures could use so many plots! But if it's characters you prefer to deal with, then Chapter 5 offers you information on everything from speech to companionship. The next chapter is bound to be useful - how to develop and test your game - here comes the "Art" of the adventure. The author quite clearly lays down these key areas to help you achieve an exciting and atmospheric game:- Clarity, Consistency, Responsiveness, Progress, Reward and Anticipation. Each one is dealt with separately, along with images, sound, text and more!

Once your adventure is written, you'll need to know how best to sell it. The book thus deals with publishing (on your own or through a company) and ways to copy-protect the game to reduce piracy. Several methods are given here, such as making the instructions essential and hard to photocopy. All good stuff. Chapter 8 follows on about the architecture of adventures - this means locations, objects and so on. The next chapter is unusual in that it offers a text adventure example, taken from the author's own game, "Sir Ramic Hobbs and the High Level Gorilla" - if you don't get the title, pronounce the second word "Rammic"! Although the example, complete with inputs and responses, covers several pages, it's an interesting insight into how your game could look, with plenty of added humour. Infact it makes you want to load and play the game yourself - what better recommendation?.

The book is completed with four appendices and an index. These cover game-writing programs, adventures the author has played and discussed in the book, a bibliography of other useful books, and finally a checklist of development stages, background themes and plot elements.

What can I say? The book is brilliant! It contains so many good ideas and useful advice that I'd go so far as to say it's an essential purchase for anyone even contemplating writing an adventure, or playtesting one! My only gripe is that it wasn't around when I wrote my first game! I could have learnt so much from it - the plot ideas alone make it worth buying. It's an attractively presented book as well, with a full-colour cover and several great illustrations dotted about inside. It really does seem as if Gil Williamson has researched every detail about adventure writing... and spilled the beans! There's simply no excuse to be without a copy.

Get yours today from:- AMAZON SYSTEMS, Merlewood, Lodge Hill Road, Farnham, Surrey, GU10 3RD. Price: £4.95 (+ £1.50 outside UK).

COMPUTER ADVENTURES
- THE SECRET ART -

This book is the first to concentrate on the Art rather than the Technology of Computer Adventures. It is a MUST for every adventure writer.

Paul Rigby in 'CRASH' said "...can be thoroughly recommended to any adventure author or anyone else who is thinking about creating their own adventure game or who has a general interest in adventure games."

Paul Glancey in 'C+VG' said "...provides a throrough analysis of the design techniques used in all the best adventures... novice RPG programmers are sure to find it invaluable."

Chris Crawford in 'Journal of Computer Game Design' reviewed it so favourably that we have had a flood of orders from the USA.

It costs just £4.95 - post free in the UK, post £1.50 Worldwide.

Send a cheque/PO to Amazon Systems, Merlewood, Lodge Hill Rd., Farnham Surrey GU10 3RD, or phone (0252) 716669 with your VISA or ACCESS.