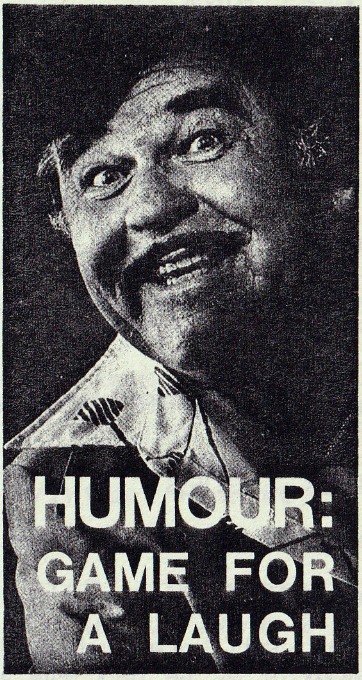
Adventure Workshop

SEAN ELLIS

AMOS

ADVSYS

TALESPIN



HUMOUR:
GAME FOR A LAUGH

Virtual Reality

ISSUE 2
£1.25

ADDRESS

The address for ADVENTURE WORKSHOP is:

3 West Lane,
Baildon,
Near Shipley,
West Yorkshire,
BD17 5HD, UK.

PRICES

You can order just one issue or as many as you like, at the costs below:

£1.25	- Britain
£1.75	- Europe
£2.25	- Elsewhere

Please note that these prices INCLUDE postage and packing.

PAYMENT

You can pay by cheque, postal order or with stamps for small amounts, but NO COINS PLEASE. Readers outside Britain should pay with International Giro cheques.

All cheques and postal orders should be made out to "C.HESTER" only, NOT the name of the magazine.

BACK ISSUES

These cost just the same as the current issue.

ADVERTIZING

You may wish to place an advert in ADVENTURE WORKSHOP. If so, the costs below are all you'll need to know:

£6	- back cover
£5	- inside full page
£2.50	- half a page
£1.25	- quarter page

Note that the back page is always on coloured paper and that you are buying the whole page - it can't be split into half or quarter, unlike the pages inside, which are always on white paper.

LETTERS

Should you wish to write in to me, I'll be only too glad to read your letters. However, if you want a personal reply outside of the magazine, please enclose a stamped and addressed envelope. Thank you. If you don't want me to publish your letter in ADVENTURE WORKSHOP you must write "NOT FOR PUBLICATION" on the top.

FORMAT

This is an amateur magazine produced on an irregular basis as a non-profit hobby.

PRODUCTION

I design the whole magazine and paste up the artwork. Mandy Rodrigues (of ADVENTURE PROBE magazine) then publishes it with the help of her husband John.

COPYRIGHT

ADVENTURE WORKSHOP is (C) 1991 Christopher Hester (the Editor) except for contributions where the author retains full copyright on their material, and are free to use it elsewhere.

CONTRIBUTIONS

These are always welcome. I'd prefer your articles ready typed with a small margin around them, Make sure the ink is as dark as possible so the text will photocopy clearly. Handwritten stuff is also welcome! Drawings, cartoons and so on are also welcome, so send in whatever you can.

Finally, thank you all for seeking out this magazine and for taking the time to read it.

Yours, Christopher.

Chris

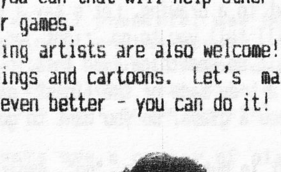
Sean Ellis has kindly offered to help users of Incentive's "Sixteen Thirty-two Adventure Creator", better known as STAC. If you've anything you'd like to know about the internal workings of the program, or anything else you can't fathom out about STAC, why not write to me and I'll pass your letters on to Sean. (First though, read his letter on Page 31.) You're sure to get a good answer - Sean Ellis wrote STAC!

Virtual Reality is of great interest at the moment to game designers who might one day produce the ultimate adventure with you actually appearing inside the game! Following William Gibson's prize-winning debut novel "Neuromancer", constant efforts have been made to replicate his vision of a computerized world you don't just watch on a screen, you think you're there! Progress is slow but the potential (especially for realistic adventures) is immense. Or is it? I(a)nspector Eveleigh investigates on Page 24...

For beginners, Steve Clay is at hand with a couple of neat articles on writing humour in adventures and other tips. For experts, Gerald Kellett and others have plenty more to offer on machine code and how to make full use of the best utilities around at the moment. There's something for everyone!

If not... why not tell me? I'm open to suggestions about improving the magazine. I'm also keen to see a lot more articles come in. You don't have to be a genius - just send me anything you can that will help other writers with their games.

Budding artists are also welcome! Send me your drawings and cartoons. Let's make this magazine even better - you can do it!



The Tales of Talespin

by Brian Pell

Here I am again going on and on about Talespin. I left you last time writing about ways of getting around the program to get some sort of adventure. This time I'm going to write about how to use the program itself. To start any game you must have a beginning, so with Talespin your first page must be called "start". Talespin uses pages instead of locations so for now on I'll use the word "page", it's also shorter to type than "location". The start page can be your first location or your title. If you use it as a title page, try to design it by using pictures you're going to use in the adventure - it adds to the game and also saves on memory. By now you should have your adventure all worked out on paper and the drawings roughly sketched out. So we have a title page and a start to your mega game. The only drawback I found with having the start page your first location was you need a page where you want to set your values to zero, ie: to see if an object is in a place or whether it's dark or so on. If you have the start as a location then everytime you enter it the value on the page entry restarts. So what I have done is to have a title page and on it set all the values to their start positions and only entered it when the game was restarted. Of course some pages you may want to restart a value, ie: a monster will reappear or something. One good tip I found about the drawings was to draw them large so you can shrink them down to use to place in the distance. Talespin can shrink things down but can't blow them up. To save space with your drawings plan them out, don't draw a tree with leaves on - draw it bare then draw a bush. You can have three different pictures which can be used for many scenes, ie: a bare tree or a bush and also use the bush for the leaves on the tree.

If you want a maze in a forest you can get a difficult one by using only two pages, like this:- call each page say, scene one and two. Then set a variable say, 'maze' zero to ten. On page one place the trees and bushes where you want and give them a condition of zero. This means that when the variable is zero on scene one the pictures are drawn. The condition must be put into the main menu on the condition of drawing. Go to the page entry and set it to zero, so everytime you enter, the page is set to zero. We'll only use it here for the moment to make up the page - when the game is finished I'll remove it. Still on the same page change the page entry to one and restart the page, click on normal and all your work will disappear, but don't worry it's only because the drawings you've just put on the page will only appear when it's zero, it's now set to one. Place some more drawings and condition them to one. You now have two scenes on one page. You can do this until you run out of space on the page. If you do the same to page two, you now have about eight or nine locations on just two pages. Now let's get them to work and move from one to another. Go to page one and set the page entry to zero - all the drawings appear. Go to the text operations and add some text to a drawing. Let's say a tree on the right will tell you to go right or east. The text could be something like this:-

"You see a tree made of the finest oak and beyond you see a glade. Do you want to go east yes or no."

Type the text in but add the "no" separately, you'll see why in a minute. Make a variable called 'movement', let's say, and give it two or three values. Make movement zero on all the text, but for the "no". Set that to one and add a 'goto page two' if you want to enter another page or set scene to another value and it will change the drawings. So you have moved to another scene on another page or still on the same page but the drawings have changed. The text will only appear when the tree appears.

A Whole STAC Of Problems

By Matthew Conway

Part The Second: What You See Is What You Get

Right then, you've read my article on getting the most from Special Condition 17 and so now your adventure opens with a beautifully-designed title page which draws the player into the game immediately. He marvels at its aesthetic value, reads the on-screen messages, presses a key to access the adventure proper and...

Good god! What's that? The artistic title page vanishes to reveal a poorly-presented, semi-illegible location description which immediately destroys the captivating atmosphere you strived to attain. The player's estimation of your adventure suddenly drops by 1000% and he wonders if he hasn't bought a turkey after all.

Well, okay, maybe a slight exaggeration there, but the point remains valid. Having slaved away to produce a perfect introduction, there's nothing worse than showing your slackness by leaving the player with a first location description which leaves a lot to be desired. The point of this article is to try to help you avoid this easily-found pitfall and instead give your adventure a much better presentational look.

In fact, this can be achieved very easily with what are, generally, only small changes. There is one large routine, which I'll show you in a moment, but even this is relatively straightforward and easy to implement.

The shorts first, though:

1) Always make sure that you place a clear-screen character at the start of room descriptions, both long and short. This is achieved by pressing Shift-Help followed by Shift-Home and results in the locations being printed at the top of a blank screen - much neater than having to search through piles of text just to find where you are.

2) Similarly, always add a Control-J followed by Shift-Help and Return at the start of the You can also see message because this makes the list of available objects more conspicuous and saves the player having to search through the end of the location description for it.

3) If at all possible, use 80-column text because this halves the depth of any text printed on the screen, consequently making it much neater. It also cuts down on the amount of untidy scrolling which takes place. However, it's better to use 40-column text with a legible font than 80-column text with an illegible font so be careful.

4) Always use a cursor of size 7, ie: don't tamper with the cursor command: a solid square is easier to follow than a thin line.

5) Finally, always check the message or location text you have entered by seeing what it will look like in the actual adventure. STAC can and will print brackets and quotes at the end of one line and the text they enclose at the start of the next and nothing can be worse than seeing this type of error. If necessary, juggle with the text so that everything appears on the same line.

Right, now that that's over, I'll finally reveal the routine which I promised you earlier in this article and also in the last issue of Adventure Coder. It is quite long-winded for what it does, but it's simple enough when you think about it and it does the job well so I'm sticking with it! Anway, what it does is to smarten up the results of the list command. Normally, this just prints all the objects in a room with commas between each pair and leaves you to tack a full-stop on at the end. Unfortunately, things like You can also see a sword, a shield, a lamp. don't smack of particularly good English. Wouldn't it be nicer if, instead, the message on the screen was You can also see a sword, a shield and a lamp. and the full-stop was automatically added? Well, here's the routine that does just that.

```
Messages
1       .
2       and
3       ,
9913 (DOWN)( CR )You can also see
9916 You are carrying
9917 nothing.
9927 It's pitch black. You can't see a thing.

Low Priority Conditions
        if verb "i" then special 19 wait

Special Conditions
17      1 mess$ 1 2 mess$ 2 3 mess$ 3 added to the start
14      if set? 1 and reset? 2 then message 9927 return
        descing room
        if zero? firstob room then return
        repeat
        firstob room to 9999
        until zero? firstob room
        message 9913
        repeat
        ( ( firstob 9999 ) + 9000 ) mess$ 0
        if cntobj 9999 = 1 then 1 add$ 0
        if cntobj 9999 = 2 then 2 add$ 0
        if cntobj 9999 > 2 then 3 add$ 0
        print$ 0
        firstob 9999 to room
        until zero? firstob room
15      if set? 1 and reset? 2 then message 9927 return
        set 0
        if visit? then descsht room draw pictof room else descing room draw
            pictof room
        visit
        as for Special Condition 14 from the first 'repeat' to the end
19      if zero? firstob with then message 9916 message 9917 wait
        as for Special Condition 14 from the first 'repeat' to the end except
            that all references to 'room' should be changed to 'with', the
            reference to 'message 9913' should be changed to 'message 9916' and
            a 'wait' should be added at the end
```

Phew! That isn't quite it yet, though. In addition to the above, you'll also have to enter the short object descriptions into the message table as well as into the object table: these should occupy positions 9000 greater than the object number, so if object 1's short description is a sword then message 9001 would also be a sword and so on. Make sure that these descriptions are 30 characters long at the most, though, otherwise you'll start getting characters cut off the end!

I suppose you want to know how all this works? Oh, very well then. The addition to Special Condition 17 sets up strings 1, 2 and 3 to contain the same text as messages 1, 2 and 3 respectively, ready for when a room description, look command or inventory command is entered. Note that the line in Low Priority Conditions for dealing with inventory commands replaces the two given in the STAC manual and directs the whole routine to a new Special Condition, number 19, so that loops can be used.

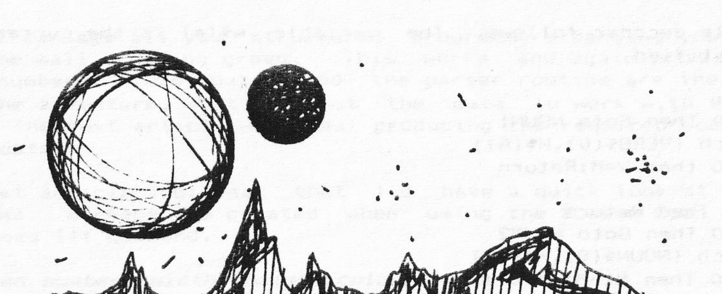
For an example, we'll look at what happens if the player types look. A Low Priority Condition passes control to Special Condition 14. If the room is dark and the player has no source of light then the message It's pitch black. You can't see a thing. is printed and that's that. If the room is lit in some way, however, the long room description is printed and the routine really begins to do its job. If there are no objects in the room then, again, that's that and the routine is exited. If there are some, however, they are all moved to room 9999 which must not be used for any other purpose whatsoever unless you like your adventures to work illogically. Then, the message You can also see is printed on a new line. What is printed on the screen next depends on how many objects are in room 9999. If there is only one then its short description is printed followed by a full-stop; if two then the short description is printed followed by and because the next object must be the last; if three or more then a comma is added. Finally, objects are moved back into the player's location one at a time until none are left and the routine ends. The same general thing happens when a room is described upon entry or an inventory is acted upon.

Looking at the listing, you may wonder why it is necessary to bother with strings at all and not simply tack messages onto the end of object descriptions. This is because STAC's willingness to put commas at the start of lines rears its ugly head and the only way to combat this is by placing the object description and the punctuation in the same message, hence the add$ which does just this.

By the time you've typed all that in, you may think that the result is pretty minimal and, ordinarily, I would agree with you. However, when you consider what you can do with STAC and the unlimited memory you can make use of due to being able to link files, it should be obvious that it is the little things which add together to make a big difference. There's no reason why you shouldn't try to tidy up all of STAC's inadequacies and, in doing so, you may stumble across a piece of code which proves to be invaluable. I have on many occasions and I hope to able to share a few more of them with you in the future.

Anyway, that's it for this article. If any of you out there have found a quicker way of doing what this article does, send it to me immediately! It could save a lot of people a lot of typing and sanity... Of course, if anybody wants to get in contact with me because they have a useful routine, a question or merely a comment to pass on, please do. Just remember that an SSAE is necessary if you want a reply outside of these pages.

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



AMOS Avenue

By John Ferris

This article concerns the "decoder", the name I have given to the routine which picks out valid verbs, nouns and adverbs from the command.

From the last article, the parser produced an array of words called W$(). Lets imagine that the command was "KILL DWARF WITH SWORD". This gives a W$ array of four elements. There are two ways of searching for these words, the tidy way, or the messy way!

The obvious AMOS command is Match, in the format

```
                    M=Match(VERB$(0),W$(X)).
```

If a match is found, M will contain the index number, ie the element number
of VERB$ that is identical to W$(X). If no match is found, M is negative.
The array VERB$ has to be sorted using Sort VERB$(0) before Matching. Now this means that your list of verbs (along with nouns and adverbs) have to be in alphabetical order to begin with. This is because the array will be sorted as follows:

```
        1  b            to                    1 a
        2  d                                  2 b
        3  a                                  3 c
        4  c                                  4 d
```

and so "b" is changed from element 1 to element 2. This means VERB$(1) is no longer "get" but "attack" with obvious problems. Good programmers (I exclude myself) will have already written down every verb, noun and adverb to be used before turning the computer on. Therefore they can simply write a program which inputs their verbs, sorts them then saves the sorted array to disk. They can get verb numbers simply from a printout of the array. The problems start if you find that a verb is missing. This could mean altering large chunks of code simply because every verb number is greater by one, hence all the references to these verbs have to be changed.

A simple decoder follows. The variable w$(a) is the current word to be identified.

```
DECODE:
If V<>0 Then Goto NOUN1
M= Match (VERB$(0),W$(A))
If M >0 then V=M:Return
NOUN1:
If A=1 Then Return
If n<>0 Then Goto NOUN2
M= Match (NOUN$(0),W$(A))
If M >0 Then N1=M:return
NOUN2:
If N2<>0 Then Goto ADVERB
M= Match (NOUN$(0),W$(A))
```

```
If M >0 Then N2=M:return
ADVERB:
if AD<>0 Then Return
M= Match (ADVERB$(0),W$(A))
if M >0 Then AD=M:Return
Return
```

Although I have not tested this routine it should work, as long as the arrays VERB$, NOUN$ and ADVERB$ are sorted. The decoder produces the variables V, N1, N2 and AD which contain verb, noun and adverb numbers which we can work with.

This is all well if you already know every word you are going to use. But what if, like me, you don't? Well, you can't use the Match statement for a start. The following routine is an alternative decoder, ungainly and messy but functional. Each array has, for example, 10 elements in it. In practice, they will each be larger.

```
DECODE:
If V<>0 Then Goto NOUN1
CC=0
While CC<=10 And V=0
Inc CC:if VERB$(CC)=W$(A) Then V=CC
Wend
If V<>0 Then Return
NOUN1:
If N1<>0 Then Goto NOUN2
CC=0
While CC<=10 And N1=0
Inc CC:If NOUN$(CC)=W$(A) Then N1=CC
Wend
If N1<>0 Then Return
NOUN2:
If N2<>0 Then Goto ADVERB
CC=0
While CC<=10 And N2=0
Inc CC:If NOUN$(CC)=W$(A) Then N2=CC
Wend
If N2<>0 Then Return
ADVERB:
If AD<>0 Then RETURN
CC=0
While CC<=10 And AD=0
Inc CC:If ADVERB$(CC)=W$(A) Then AD=CC
Wend
Return
```

Alright, I can see all you structured programmers banging your heads against the wall or going green. This works and again produces the required numbers. This routine and the parser routine are the engine room of the adventure, but without the data to work with they are useless. The next article will deal producing the required vocabulary and text data.

In the last article I did say that I'd have a quick look at screens and windows. Screens are created when using the Screen Open command and the Load Iff command.

Screen Open *number, width, height,colours,mode*

Number is the screen i.d. from 0 to 7 (i.e. a total of eight screens displayed at once!)

Width, Height are the size of the screen. A screen can be a small square in the middle of the display or a huge map many TV displays wide and high.

Colours sets the number of colours available to the screen, from 2 to 32, with 64 for Extra Half bright mode, 4096 for HAM!

Mode is either Lowres or Hires.

Load Iff "*file*" [,*screennumber*]

Loads an IFF graphic "*file*" into the present screen or screen *screennumber*. If *screennumber* doesn't exist, then it will be created and the file loaded in. If it does exist, then the incoming data will erase the current contents of the screen.

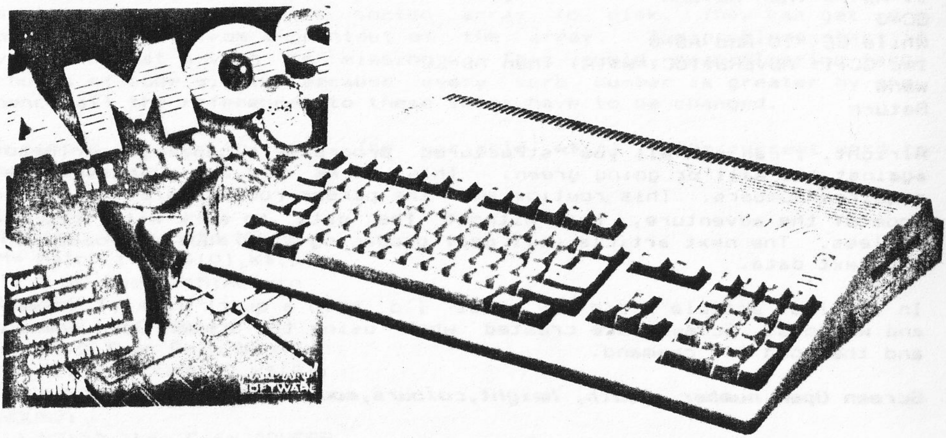
Windopen *number,x,y,charwidth, charheight*,[*,border*]

This opens a window in the current screen. *Number* is the i.d., *x* and *y* are the top left corner coordinates of the window. *Charwidth* and *charheight* are the width and height, in text characters, of the window. Setting these to 70 and 10 produces a 70x10 text display. Finally *border* is the style of the window border.

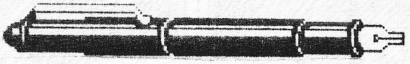
Using these two commands it is easy to produce a split screen display for both text and graphic adventures. Try loading up an art package, Dpaint will do. Paint a picture covering the top third of the display, remembering to turn off the side panel to get at the full width of the screen. Now create a brush using the top third of your picture and save this as a brush, ie "brush". Load AMOS and type this in direct mode:

```
Load Iff "brush",1
```

Your graphic will appear across the top of the screen. Putting graphics into a game is as simple as that. The hard bit is drawing GOOD graphics!



WRITE ON



by Steve Clay

This article is an attempt to provide a little help on the messy business of writing style, an area of game development that is a true reflection on the writer's ability.

Tense
(Relax in a Radox bath)

As in past, present or future. Well future is out the window by its very nature: 'You will go north and see something strange' ermm.. Past tense gives a storytelling feel to a game: 'I went north and saw something strange'. Present tense gives a rigid feel to the game: 'I go north, I see something strange'.

Style
(Don't wear flares, they burn your legs)

Will your game be a side-splitter or a stomach churner, will the computer act as the hero receiving instructions from the player or will the player adopt another role.

'Upon examining the skull I found it attached to a skeleton, which judging by the looks of it had overdone the hip and thigh diet.'

'The bone of the skull was deathly cold, small flecks of red light burned in the eye sockets. From the corner of the grinning mouth a beetle crawled.'

'I open the door. There is a skull here. I examine it and as my fingers touch the bone the jaw springs open. Inside is a key.'

'Big Ralph walked toward the skull. As he approached, the eye sockets flared with bright red light.'

Locations

These are hopefully what give your game an air of existence, they make your game-world real. The compromises however must start here and you must make your locations as real as possible in as short a space as possible. Chris wrote a piece on atmosphere in an earlier issue and is well worth a read. My examples will be of a 'Small cave with a passage leading north'. I have written the description of the same location in differing styles and varying degrees of atmosphere.

'I stood at the centre of small cave, to the north a passage, cut into the harsh mountain rock, lead into a shallow-filled gloom.

'The walls of the cave are a mass of rude and suggestive phrases, above the northern archway some idiot has fixed a sign which reads PLEASE DON'T REMAIN STANDING ONCE YOU ARE DEAD AS THE JANITOR HAS ENOUGH TO DO ALREADY!'

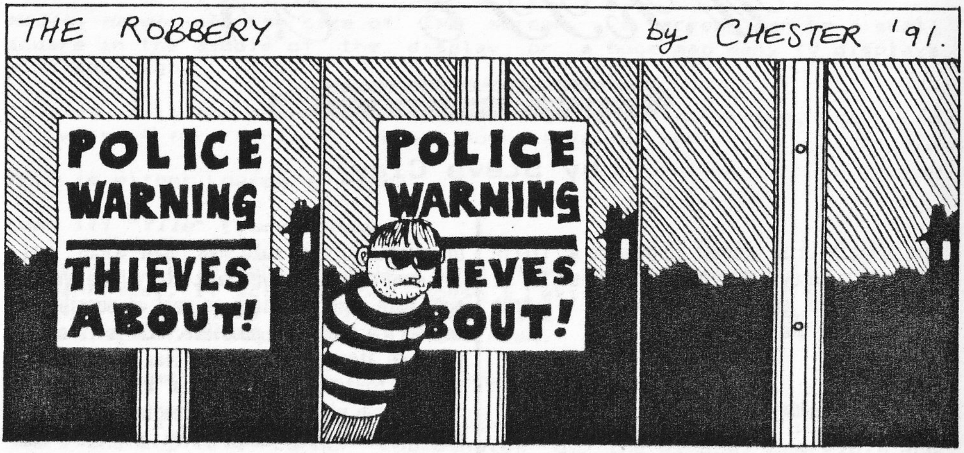
'I feel a cold, deathly chill take hold of my body, as if the rock itself is trying to enter my soul, through a small unwelcoming hole in the north wall shadows move to and fro.'

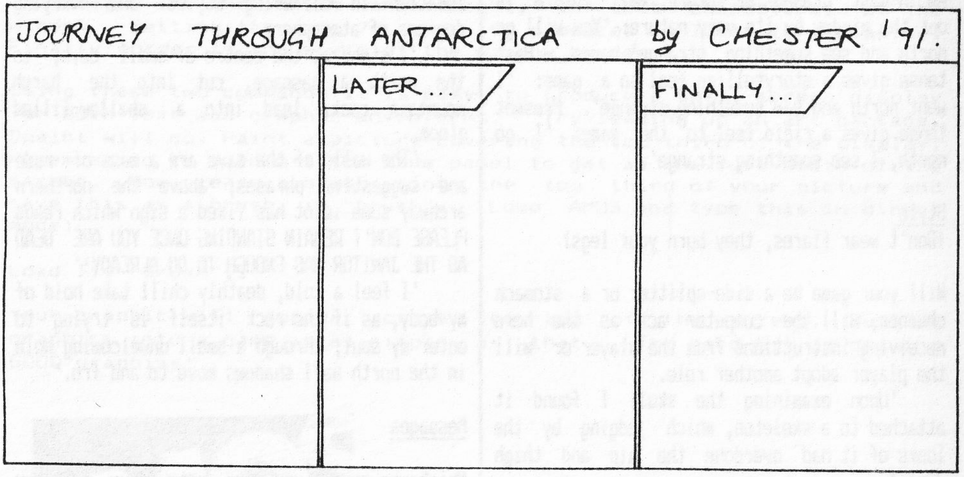
Messages

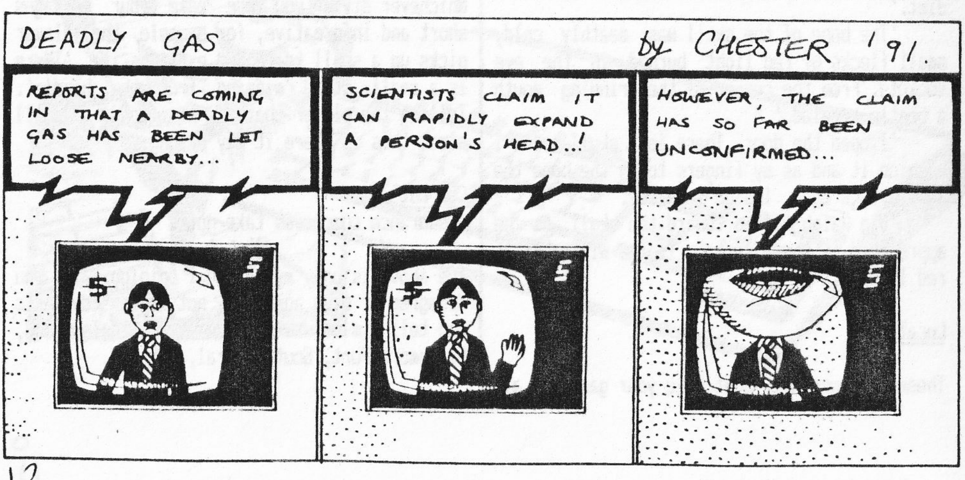
Whichever style you use make your messages short and informative, for example the player picks up a small knife and examines it: 'There is a small stone missing from the handle'. Telling the player that if he finds a small stone this is where it may belong.

Cop-out
(Insurance companies take note)

The above is only my personal opinion. If you disagree or even agree why not write to Chris and tell him or write to me at 17 Stanlaw Road, Ellesmere Port, South Wirral, L65 OEY.


THE ROBBERY
by CHESTER '91.
POLICE WARNING
THIEVES ABOUT!
POLICE WARNING
IEVES BOUT!


JOURNEY THROUGH ANTARCTICA
by CHESTER '91.
LATER...
FINALLY...


DEADLY GAS
by CHESTER '91
REPORTS ARE COMING IN THAT A DEADLY GAS HAS BEEN LET LOOSE NEARBY...
SCIENTISTS CLAIM IT CAN RAPIDLY EXPAND A PERSON'S HEAD...!
HOWEVER, THE CLAIM HAS SO FAR BEEN UNCONFIRMED...
5
5
5

HEAD'S CASES 2: Dead men don't wear lace

Ritchie's 'da name, Ritchie Head, private h'investigata' fa' hire, no one wants ta' gimme' a proper job anymore, kinda' like 'da stores do fa' 'da kids when 'da chris'mas sale's on..

It wuzza' slow kinda' mornin', so slow 'dat even 'da grannies wuz' muggin' each udda' ta' pass 'da time away!
I grinned over at ma' new secretary, Wendy Day, as I lifted ma' feet off ah' 'da desk an' sifted t'rough 'da sports pages ah' 'da mornin' newsrag, she'd bin' wid' me fa' more 'dan t'ree months now - 'da secretary not 'da paper, I've hoid uv' old news, bud' 'dat's ridiculas! - bud' I wuz' still havin' trouble tellin' 'er an' 'er 2 sisters apart, Mandy an' May..
Suddenly 'dere wuz' a rap at 'da door - bud' 'da las' t'ing I needed right now wuz' t'ree black guys singin' in rhyme - So I grabbed 'da note I ah' 'dem 'anded me an' I told 'em 'dis wuz' a defective agency not MTV, so I slams 'da door an' sits back down again, 'da note said.. Well, it didn't say anyt'in' really, it wuz' only a bit ah' paper, so I read it instead..

'Dear Mr Head,

I wish to employ your services for an unknown period of time, but I please request that you meet with my chauffer outside the Metropolitan Museum of Art at 2.30 this afternoon..

Yours sincerely

S, S, Syde

Sue. S. Syde (Ms)'

"Cheers, doll!" I sez' ta' Wendy as I grabs ma' mac' an' heads out ta' 'da bus, 'dere were tears in 'er eyes as I left 'da office, guess I must ah' slammed 'da door on 'er fingers!

Christ 'da weather down 'ere in Greenwich Village 'dis time ah' year wuz' so lousy ya' could git' all ah' four seasons in 1 day, an' maybe 'da Jackson Five if ya' wuz' really unlucky!
I grabbed an uptown Greyhound - it nearly tore ma' arm off 'da speed it wuz' goin' ya'd t'ink it wuz' still onna' track! - headin' upta' Central Park.
I got off on 57th an' headed inna' 'da park an' 'da Metro Museum - an' jus' outside 'da gates 'dis big guy shuffles over..
"Sirrr?" he growles, like a gorilla wid' toiminal piles - bud' I knew I wuz' in no real danger, like.. 'Cause 'ey looked 'bout as good in a fist fight as 'da Venus Da Milo!
"Hay!" I shouted over 'da hood 'uv 'is Limo' "You chauffer 'er ladyship's car?"
"No, sirrr!" he moanes "I'm Karr her ladyship's chauffer!"
"Ya' wiseass!" I sez' 'andin' over 'da note I 'ad "So wa' does she want?"
"Her shadyli.. Umm, her ladyship simply wishes to employ your services for the evening, to cure a problem for her!" 'da driver sneered..
"Hay! 'Da woild's fulla' guys 'dat could 'elp 'er wid' a problem like 'dat, looks like I won, huh? So does 'dis involve doin' anyt'in' unspeakable to 'er ladyship 'den?"
"No!" snapped 'da driver "Of course not, she simply wishes you to find the killer of her late husband!"
"Uhh.. S'dat all?.. Pity! So, like when does 'ey git' 'ere 'den?" I drawled smoothly, like a well oiled masseuse.
"He doesn't!" squealed 'da driver.
"Bud' ya' sez' 'ey wuz' late?"

"I mean he's dead, craphead!" screamed 'da chauffer.
"Hay! 'Da name's Ritchie, not crap, asshole! I know whut' I wuz' christened fa' christ' sake!"
"Your first name is Fukrisake?" he winced.
"Aah, shaddap' an' git' in!" I told 'im..

So I climbs inna' Limo' an' we heads up inna' 'da hills an' 'da posh area ah' town, where 'da money don't jus' grow on 'da trees, it's used as foitaliser on 'em!
'Da only info' I could git' outta' 'da driver as we gotta' 'da house, wuz' 'da old guy had a sort ah' unexpected death - especially ta' 'im! - an' 'dat 'dere wuz' no witnesses!
Geez' I had a lot ta' work on, as I squeezed outta' 'da Limo' an' headed fa' 'da house..
"Ah-ha!" I sez', grabbin' 'dis real weird guy, openin' 'da door an' tryin' ta' sneak back off inta' 'da shadows..
"Gotcha', bum!" I snarls "Sneakin' out, tryin' ta' escape, 'ay?"
"But I'm supposed to open the door, sir!" 'ey sez'.
"Yeah, why's 'dat?"
"'Cause I'm the butler, sir!"
"Yeah? Well jus' don't go away!" I shouted as I wuz' guided straight upstairs ta' meet 'er ladyship.

I knew I wuz' gettin' too old fa' 'dis kinda' game when it took me a couple ah' days ta' reach 'da top ah' 'da stairs, so I wuz' feelin' pretty lousy when I got ta' 'er ladyship's door, I'd bin' slapped t'ree times fa' tryin' already, so I lets lousy go an' she wanders back downstairs!
I pushed 'da bedroom door open an' I steps inside, 'da ol' hinges squealin' like mice bein' castrated.. Ol' lady Syde wuz' sittin' up in bed doin' 'er nails wid' a file, hell knows why? 'Dey couldn't hold any pictures up wid' no points on could 'dey?
"You 'da dame wants ta' find 'er old man's killer?" I sez'.
"Aah, yes!" she moans "But we've always had marital problems in this family!" she coughed, lookin' inna' 'da bedside mirror. Wuzza' matter, did I 'ave a face like a Swiss cheese or sumt'in'?
"Ya' know, old granny Gnott, ma' sister, always wannid' us ta' bury 'er next to 'er 'usband when she died, so we did!"
"So?" I sez' sittin' down onna' bedside cabinet near 'da dressin' table "Whadda' 'bout it?"
"Aah.." she sighed "Don't suppose he was too pleased though, he was still alive when we did it!"
"Hay!" I sez' ta' 'da old dame "Ya' t'ink youz' got problems? On average 'dere's a guy gits' run over in Times Square iv'ry 10 minutes, an' 'ey's gettin' damn sick 'uv it I can tell ya'! So ya' wonna' gimme some details 'bout ya' husband an' 'ow 'ey died?"
"God!" she moaned "I don't know 'bout John Lennon, but that joke should have been shot years ago! Well.." she sez' "He could have been poisoned but we're not quite sure, Gardner, the butler found him in the begonias - my 'usband not John Lennon - an' it could have been the way he said 'Urgh! Aaargh! I've been poisoned' that did it, but you never can tell?"
I wuz' gettin' nowhere fast wid' 'dis Syde boid - 'da castors onna' dresser wuz' movin' pretty slow too, so I left it inna' hall an' decided ta' quiz 'da rest ah' 'da staff quick a'fore anybody else got bumped off!
'Da butler wuz' avoidin' me as I looked round 'da house fa' 'da rest ah' 'da soivants, I probably did 'ave a face like a cheesy Swiss!
Ah-ha! Right nex' door ta' 'er ladyship's room I found anudda' clue.. Shep, lord Syde's ventriloquistses dog 'ad done a runner.. Shep's dog house ain't bin' shlept in! - Shlep, err, Shep'd bin' dognapped, bud' I could ah' bin' barkin' up 'da wrong tree! So I goes down ta' 'da kitchen ta' see 'dat lousy dame again..

She wuz' jus' tryin' ta' dump a can ah' weedkiller inna' 'da trash as I looks rounda' door, so I kinda' knew I 'ad 'er red 'anded - or wuz' it jus' 'da kinky rubber gloves 'dat made 'em look 'dat colour?
"Stryknine, huh?" I sez' "So ya' bumped 'da old boid off wid' ah' helluva' poison, did ya'?" I sez', lightin' a new roll-up an' takin' a slow drag..
"Aaargh!" I screamed, reachin' fa' 'da water fosette, I'll remember ta' stick 'da right end in ma' mouth nex' time!
"Well, ja' do it?" I sez' as menacin' as a wet lettuce, an' by 'da look on 'da dames mug she couldna' bin' more surprised if she'd bin' mounted by a Rhino'!
God, 'deez dames, in ma' many years as a dic' I'd seen more fake surprises 'dan I 'ad fake orgasms, or 'ad I? Ya' nivva' can tell wid' 'deez dames!
"But, umm.. 'Butler found Shep burying it in the roses, 'told me to get rid of it!" she paniced.
Well, I'd hadda' 'nuff wastin' time wid' lousy-Brown, so I headed out ta' 'da family's private graveyard at 'da back ah' 'da house ta' see if I could 'dig-up' anyt'in' new, I'd git' round ta' 'da butler soon enough!

'Da foist few graves I passes all belonged ta' 'er ladyship's cousins, 'da Shark family, Lorne Shark, Cardhu - bud' I guess 'dey jus' called 'im Card' - an' 'is sister 'Chelle - sumt'in' fishy 'dere I t'ought, bud' I moves on ta' 'da victim's grave, belongin' ta' Cecille. D. Syde..
So I leans over an' scans 'dee h'inscription onna' front ah' Cees' Syde's gravestone, it read..

'See, 'told you I was ill!'

Geez' 'dat's a helluv' an' epitath! An' 'den I seez' 'da fresh flowers, black roses! - An' a dogs Paw-Prints, tried ta' dig 'em up I guess!
"Yeah, I kinda' guess 'dis case's startin' ta' come t'gether now, Cees' ol' son.. Bud' I might 'ave ta' ask ya' a few questions later on, you bein' 'da circumstancial evidence!" I sez' ta' 'da gravestone "So, like, jus' don't go away!"
An' I kinda' reckoned 'da gardener might know a bit more ta' tell, so I wanders over ta' 'da greenhouse, an' 'dere 'ey wuz' plantin' some weeds - bud' I guess it gave 'im sumt'in' ta' do later on, diggin' 'em back up again - an' I askes 'da gardener 'bout 'dose roses..

"Oh, yes!" 'ey sez' "We did 'ave some ah' those black roses growin' 'ere, but I gave 'em to Gardner!"
"Huh, I t'ought you wuz' 'da gardener?"
"Oh, I am, I'm Butler the gardener, an' 'ees Gardner the butler, sur'.."
"Waz' 'is? Youz' guys sharin' 'da jobs roun' 'ere or sumt'in'?"
"Nah!" 'ey sez' "I told 'ey sur, 'ees Gardner the butler, an' I'm.."
"Aah, can 'da crap!" I sez' an' I wanders back ta' 'da house, I kinda' reckoned I wuz' gettin' fed more manure 'dan 'da plants wuz' round 'ere!

Lousy an' 'er ladyship were 'da only ones left in 'da house when I gits' back 'dere, 'da soivants 'ad gone! No gardener an' no Gardner!
An' like I says ta' lady Syde "Ya' can always finda' toy-boy in Palm Beach ta' 'elp ya' spend 'da h'insurance frum' Cees' - Or should 'dat 'ave bin' 'da de - Cees -d?
Bud' I kinda' figured she wouldn't 'ave ta' spend no more on doggie dinners!
Nah, I'd 'ave no trouble findin' Cees' killers - 'da butlers always gits' ya' in 'da end - Sure, I knew who it wuz' alright, it wuz' Shep, 'da talkin' dog in disguise! Yup, 'ey wuz' wearin' 'dis-guys butler's clothes!
Yeah, Shep bein' a German Shepherd I recognised 'is accent straight away!
Fancy bumpin' a guy off jus' 'cause ya' ain't in 'is will, 'dat's a woman's job! Bud' deez' canine's capers ended right 'ere, 'dis wasn't a job fa' a dic' like me, more like 'da dog-catcher!

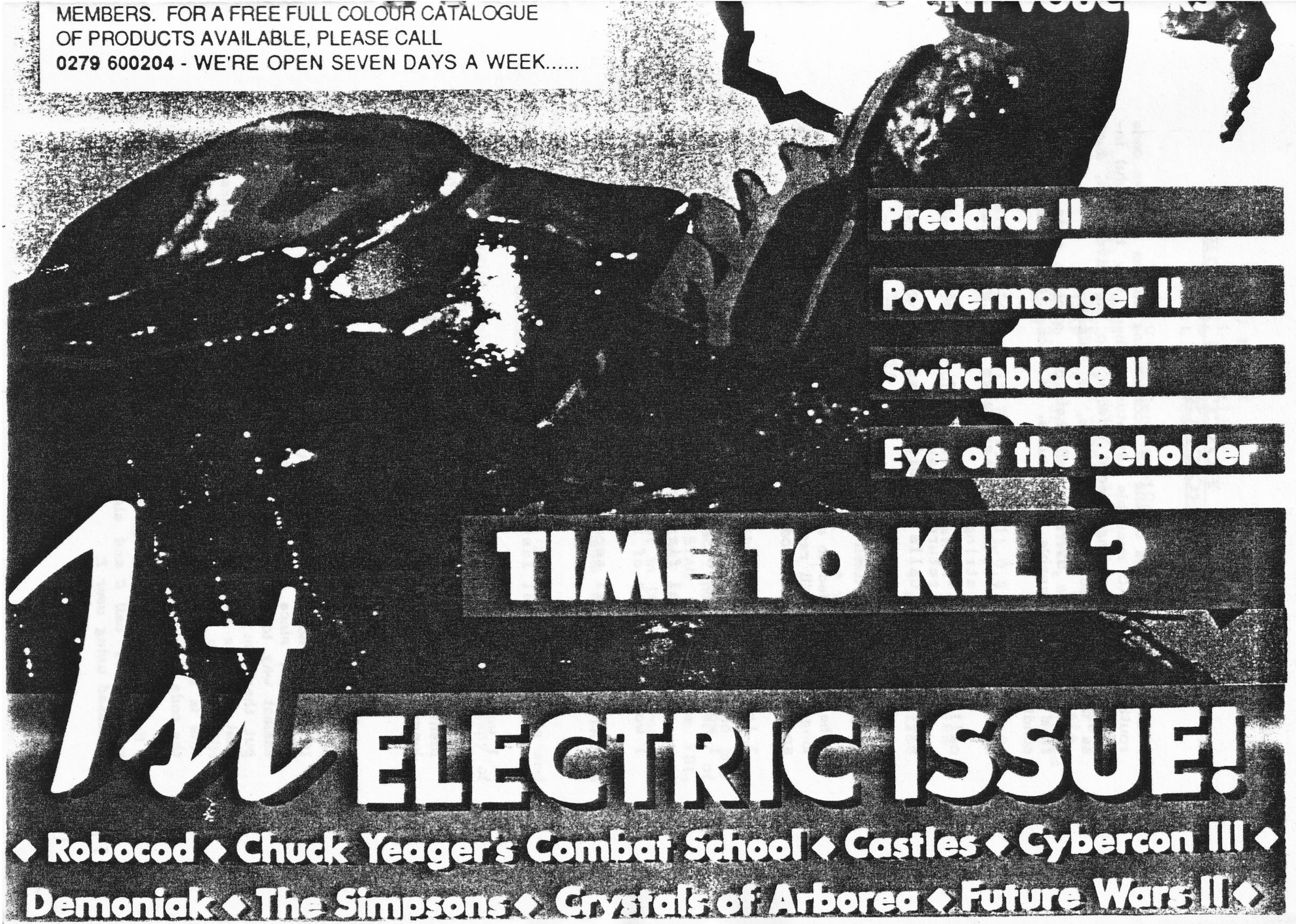
'Da end..

ISSUE 1 * JUNE/JULY 1991

For the serious 16-Bit enthusiast

THE NEW 32 PAGE MAGAZINE IN *FULL COLOUR.*
EXCLUSIVELY AVAILABLE TO MEMBERS WHO JOIN
SPECIAL RESERVE - THE LARGEST COMPUTER MAIL
ORDER COMPANY IN EUROPE, WITH OVER 50, 000

MEMBERS. FOR A FREE FULL COLOUR CATALOGUE
OF PRODUCTS AVAILABLE, PLEASE CALL
0279 600204 - WE'RE OPEN SEVEN DAYS A WEEK......


Predator II
Powermonger II
Switchblade II
Eye of the Beholder
TIME TO KILL?
1st ELECTRIC ISSUE!
◆ Robocod ◆ Chuck Yeager's Combat School ◆ Castles ◆ Cybercon III ◆
Demoniak ◆ The Simpsons ◆ Crystals of Arborea ◆ Future Wars II ◆

STAC USER ROUTINE ACCESS PROTOCOL

This document gives details on incorporating custom machine code routines into STAC runnable adventures. These may be useful in-house for incorporating special effects into the adventures, such as digitized speech, sound effects, or animation.

The STAC allows users to call their own machine code routines from within runnable adventures. It does this using three conditional words:

loadbin — Loads a binary image file whose name is in string 0.

call n — Calls a stand-alone routine that does not return a result.

user n — Calls a routine that does return a result.

These will *only* work within runnable adventures. Access to them while in the STAC itself is ignored.

The code is loaded to address $16000 and may be up to 32 Kbytes long. Thus you will require an assembler which can assemble to a specific address. The code format is as follows:

3 longs	File header
1 word	Length of first routine
	Actual first routine
1 word	Length of 2nd routine
	Actual second routine
	...
	Actual last routine
1 word	$FFFF

The header must be constructed on the front of the file to identify it to the STAC as a code file. This consists of three longwords containing the "magic numbers"

$86347954
$56794521
$32970875

If this is not present, then the STAC will refuse to load the given file, since it will not be recognized as a code file. The easiest way to get this is to start your code 12 bytes early, and put these on the front as DC.L directives. You must make sure that there is no GEMDOS file header in front of these since they MUST be the first 3 longwords in the file. The easiest way to do this is to assemble the code to the given address, then just save the code as a straight binary file dump.

Here is an example of a short routine which makes a noise, accessed by ***call 0*** and also one which returns the system time, accessed using ***user 1***:

```
                ORG $15FF4
                DC.L $86347954
                DC.L $56794521
                DC.L $32970875

                DC.W rout0end-rout0start * len of routine 0

rout0start      MOVE.L #soundata,-(SP)   * actual routine
                MOVE.W #32,-(SP)         * makes a noise
                TRAP #14                 * note that data
                ADDQ.L #6,SP             * is also included
                RTS

soundata        DC.B 6,15,7,$F7,12,4,11,0,8,16,13,0,$FF,0
                EVEN

rout0end        DC.W rout1end-rout1start * len of routine 1

rout1start      MOVE.W #23,-(SP)         * get system time
                TRAP #14                 * in D0, which is
                ADDQ.L #2,SP             * where we want
                RTS                      * the returned
                                         * value

rout1end        DC.W $FFFF               * end of list
```

The routines have full access to the operating system, and may corrupt all registers except the stack pointers. Beware of altering interrupt vectors, since the STAC revectors the VBL, keyboard, and some of the MFP interrupts through time critical routines. Returned values are passed out in D0, and the routines are terminated using an RTS opcode.

The reason that these are not available within the STAC is twofold:

a) The space used is also used by the STAC editor routines and would thus become corrupted.

b) The routine could easily be used to save a runnable copy of the STAC which could then be used for piracy purposes. There is no fear of this in the runnable adventures since the code is already unprotected.

Sean Ellis

ADVSYS: a review

by George March

I recently purchased 'Double Adventure Disk 3' from Titan Games (as advertised in 'Adventure Probe' vol' 3 no' 11 page 22), for my ST, containing 5 games and an adventure creator named the 'Adventure Writing System', the games themselves when you compare them to Linda (Marlin Games) Wright's ST version of 'Jade Stone' written with STAC, look and play an awful lot worse than many a Quilled or PAWed Speccie game I've seen in a hell of a long time, even though I only paid £3.50 for the disk! So if you really want a budget ST game, try Jade Stone, or maybe Jack (River Software) A. Lockerbie's 'Hammer of Grimmold', etc..

But anyway back to this review of the 'Adventure Writing System' (otherwise known as ADVSYS, have a look at the 'Utilities Available' column in issues 4 and 5, or do we have 1 in this issue, Chris?), now if you're a 'Probe' reader, do you remember John Packham's review of AD.LAN from vol' 3 issue 8? If not, how about his 'AD.LAN Adventuring' column in 'Coder', because ADVSYS appears to be set out in much the same way, with '[' and ']' shaped brackets for otional phrases, '(' and ')' brackets, star-shapes, semi-colons and '@' shapes, and funny command names, including, routines for some reason are termed 'Statements' or 'Phrases'!

Another thing is, there are no written instructions (all instructions are on the disk, so if you want to be able to look back on any instructions whilst using ADVSYS, then you'll have to hand write them yourself, which takes a hell of a long time to do, I can tell you, unless you've got a typewriter or printer!), the phraseology of the instructions is also pretty abysmal! Using terms like DEFINE to mean a simple LET action, and instead of having a process such as PAW's object-words, or an ABILITY command, you have to use commands like..

(PROPERTY 'weight value'*) with 'property' being the name of the obj', (DEFINE max-load 100) which simply gives the maximum weight a player can hold, so why didn't they just use a STRENGTH 100 command to make it easier? Also ARTICLE (which is basically a pronoun for an obj', like IT, THAT or THE, etc), ARG-NAME, &AUX TMP-NAME, etc, I mean a creator like GAC, PAW, STAC and Quill, etc, ie, any creator worth investigating, has all its abilities laid out as named commands like GET, DROP, GOTO and mathimatical expressions such as EQ 'flag' 'number', or SAME 'flag no' 1' 'flag no' 2', CHANCE ?, CTR (0) + 2 CSET 0, etc, but not ADVSYS, it's as though Titan Games (or the ADVSYS writer specifically) have deliberately written the most awkward, hard to understand routines, with terrible commands, this is a classic example from page 1 of the 'instructions'..

```
( DEFINE ( Factorial n )
  ( IF ( < n 2
      1
        (* n ( Factorial ( - n
 1 ))))
```

Now the instructions for ADVSYS are set into different groups, for objects, directions, vocab', etc, including an 'Object-oriented routines package', starting with 'PROPERTY DEFINITIONS', which lists normal directions like north, south, east, west, up and down, but won't allow diagonals such as northeast, southwest, and enter/leave, in, out or jump, etc, aren't allowed either, at least not in any instructions (very funny?), but does allow commands such as INITIAL LOCATION (ie, the room no' an obj' first starts out in), and brief and verbose location descriptions are also catered for which is good, but as far as weird names go, how about PARENT (a parent of an obj'? Well that's what the not very explanitory and sparse instructions say), SIBLING (the next sibling to an object, to which I thought, 'I know what a sibling is, but what the hell's it got to do with this?') and a CHILD command (ie, the first 'child' of an object, which is, umm.. Err?) well you get the idea, it's a bloody hard and awkward system to understand let alone use!

Now even if it does sound as though I'm panning ADVSYS something rotten, I'm not really because it does have 1 or 2 thing going for it, such as..

'Portal Properties', door commands basically! Which include CLOSED (ie, is a door closed, if this is NOT true, then the door is open, obviously!), LOCKED, KEY (ie, to open a door with, but the instructions give no idea how to implenent 1?) and a command called OTHER-SIDE (all the instructions say about it is 'the other portal is a pair', to whicn I thought 'what OTHER portal?')..

VOCABULARY DEFINITIONS are next described, extremely briefly mind! Which include..

Synonyms of words and defining the basic vocabulary, and VARIABLE DEFINITIONS which are commands such as CURLOC (ie, the players CURrent-LOCation!), %ACTOR, %DOBJECT (the direct object) and %IOBJECT (an in-direct object).

Now seriously, by the time I got to CONNECTION PRIMITIVES (which was only 3 pages down on-screen!) I was nearly climbing the walls with frustration! The whole trouble with the sparse instructions for ADVSYS is that they tell you what to do, but not how to do it! It tells you (very briefly) what a command does, but not how or why it does it!

I mean under the page entitled CONNECTION PRIMITIVES, 1 thing it says about objects is, and this is exactly how it looks on-screen!

```
: Connect all objects to their initial parents
( define connect-all &aux obj maxpl par)
  (setq obj 1)
  (setq maxpl (+ Socount 1))
  (while (<obj maxpl)
   (if (setq pur (getp obj initial-location))
      (Connect par obj))
   (setq obj (+ obj 1))))
```

That's it! No instructions telling you what this bit or that bit does, it just shows you an example! Horrendous!

Now if like me you can't tell what the hell's what in that lot above, just take a gander at this lot for LOOKing inside of a box for example..

```
(define (print-contents obj prop &aux desc)
   (setq obj (getp obj child))
   (while obj
     (if (setq desc (getp obj prop))
        (progn
          (print " ")
          (print desc)))
     (setq obj (getp obj sibling))))
```

Now if you compare that gibberish above to the same thing done by PAW, you get..

```
LOOK BOX PREP INSIDE PRESENT 'box' LET 51 'box no' SYSMESS 'The _
contains ' LISTAT 'inside of box' DONE
```

Easy peezie, everything's laid-out for you, a command for each and every action needed, and at least it's written in plain, easy to understand English!

Now to start off with, in writing anything for your ST using ADVSYS, you need to write an 'adventure description' (which is, according to these here instructions, 'an ordinary ASCII text file' containing definitions for all objects, actions, messages, location descriptions needed, etc, used as input to the adventure 'compiler' or ADVCOM, which takes the ASCII descriptions and compiles them into a set of 'data structures', to be used by the interpreter, known as ADVINT), and so, unlike AD.LAN, GAC, STAC or PAW, etc, you cannot create a playable game using ADVSYS on its own, as ADVSYS is not so much a 'creator' as just a compiler and interpreter package! Now as I said, the instructions say that first of all you need to write an ASCII file before you can do anything at all, now call me a dope, call me a thickie (which many people probably do anyway!), but in all my years of using the Quill (thanx very much, Chris!), GAC (when I had my C64) and PAW to write with, I've never had the need to come across such a thing! Ie, what the hell is an ASCII file? I do know that it's a number code representing all the letters, numbers and symbols a keyboard has, I'm not that thick (even though I do make lots of spelling mistakes and things, but how the hell do I write a file?).

Well anyway, the parser for ADVSYS seems to be pretty ok, with all the standard Verbs, Nouns, Prepositions, etc, and also includes commands like Direct and In-direct object names, etc, but does persist in having names like %ACTOR (a noun phrase), PROPERTY (which is just the name of an obj'), etc, when easier to remember names could be used!

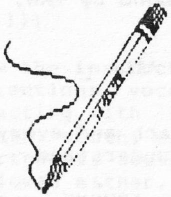
There's also no facility for error or default messages (you also have to define them from scratch, even though any good creator would have them in-built but giving the writer a facility to be able to alter the wording! There's no compressor either, and the really bad bummer of it all, has to be that there's no graphics package in ADVSYS, but I suppose you lot out there that know how to programme properly on the ST will know how to import graphics from external art-packages, so that may not be a problem and I do reckon that anyone who has come across AD.LAN or does understand ST programming somewhat, shouldn't really have all that much of a bad time with ADVSYS, and for only £3.50 for a creator AND 5 text games, it might be worth your time after all! But for me, I think STAC or an ST PAW is more my style!

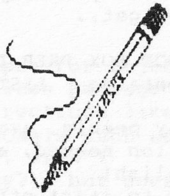
Poet's Corner

Both poems written by Andrew Green.

<u>TCP</u>

TCP - I give you my skin,
Come sting away this dreaded spot,
Your big brown bottle tall & thin,
Come cleanse this body -
It's the only one I've got.





<u>Jack & Jill</u>

Jack and Jill
Went up the hill
To fetch a pail of water.
Jack stripped down,
Took off Jill's gown,
And now they have a daughter.

GAME * FOR * A * LAUGH

by Steve Clay

Humour in adventures is often tried yet seldom succesful. Infact to try and make a game totally humourous is a very difficult task and should not be undertaken without a good deal of forethought. So how to keep the player smiling without constructing a comic masterpiece?

Names

The use of funny names could be said to be synonymous with adventures, ie Gzorp the Barbarian, Twizlpoop the Wizard, etc. So to cause a chuckle the writer must look beyond the unpronounceable and find a laugh for the right reason (because it's funny). Try everyday words for names of characters, places etc. Such as Koatanger, Odsok, Bukend and so on.

Messages

Even the most serious adventure can use humour in the response to inputs from the player, even if it is a sarcastic reply to a stupid request. Imagine the player has just battled their way through to the evil Necromancer's lair. Finding a decapitated hand, the player types EXAMINE HAND, the response could be "It's alright, it's 'armless!"

Problems

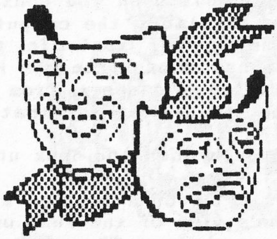
This is a difficult area in which to use humour but check out Level 9 games for clever examples, ie the cave underground that needs illumination and when an octopus appears you use it, as many hands make light work.

Locations

As Chris pointed out in Issue 5 of "Adventure Coder", use of words is vital, when trying to be humourous in your descriptions the ability to use one or two words instead of a paragraph is paramount. Example: the room had the attraction of a perspiration stain on an orc's leggings. The player now knows that this is not the rave joint in town. The use of funny names as place names is an obvious idea. (See NAMES)

Graphics

These can be made humourous by using cartoon type characters or jokes in the background, such as a passing bird with a jetpac on or similar. With graphics, however, memory is eaten up rapidly and you must decide how badly you want a laugh.



"*Even the most serious adventure can use humour* "

Finally

Never crack a joke in the game and then try and explain it. It not only kills the joke, but is also boring for the player.

Comments welcome: Steve Clay, 17 Stanlaw Road, Ellesmere Port, South Wirral, L65 OEY.

VIRTUAL REALITY

by Ian Eveleigh

So, you've seen all the hype about Virtual Reality, right? You strap this pregnant pair of binoculars to your head, try to forget that people are watching you making a complete prat of yourself and slip into another world.

At the recent European Computer Trade Show I was fortunate enough to be able to have a go. After watching several people crouching down and turning aimlessly about with big grins on their faces I decided it was time to experience this for myself.

So I slipped on the headset and looked at the world... Aha! It appears that everyone in Virtual Reality world has bad eyes! No, seriously, you know those pictures you've seen on TV of that red room with the yellow chequered floor and the floating green teapot? Well the real thing ain't that sharp. It seems to be slightly blurred by the lenses inside the headset.

Anyway, after the last victim had the teapot dropped on her head. I was given a new "world" to look at. So, I'm in a maroon room with a blue chequered floor and there's a chair and a table with two wine bottles on it. Let's look around... As you've probably seen or heard, the picture update is a tad on the slow side. But when you consider the calculations that need to be done that's understandable: the computer must first calculate where you are in the room by finding your (real-life) position and height relative to a stand which receives transmissions from the headset. It must then find the angle of the view (calculated, I suspect, from mercury switches in the headset). It must then create the display for that view. I was tempted to try a hand-stand.

"You can crouch down and look under the table if you like," I was told.

Now there's a bit of a problem with Virtual Reality: If you completely black out someone's view of the real world, and put a heavy-ish weight on their head, and a couple of TV sets in front of their eyes with slowly updated pictures of a false world you begin to lose your sense of balance! Crouching down is a real task! You're not sure if you're going to fall over or what!

The whole experience is strange, but interesting.

We're supposed to have Virtual Reality in the home sometime next year. So let's suppose the price of bacon doesn't go up and it happens. So what? What exactly are you going to do with it. Playing with teapots might get a bit boring after a few days.

You won't actually be able to walk around in this new world - you'll be restricted by the wires and (more importantly) the space available at home. So you'll be sat down using a joystick to control your movements. This basically means the only advantage of Virtual Reality will be that you'll have a headset on that allows you to look fully around. But you'll be shut off from the rest of the world, unable to share the experience with other people. And will prolonged use give you neck-aches from that weight and head-aches from those TVs right in front of your eyes?

I can't help asking myself if Virtual Reality is anything more than a gimmick? Sure, it's nice, but its uses seem limited. Think: of all the games you play, in how many of them would you actually like to "be there"? In most arcade games you watch your character run around the screen - so it's useless for that. Most adventures are enjoyable because of the written word and use interaction through the keyboard - useless. Which basically leaves you with two areas of use: simulation and exploration.

Simulations would be improved using Virtual Reality - you'd be able to look all around you in the air (or whatever) or down at the controls, but would it really be such a vast improvement? The graphics would still be slower or less detailed than you've become used to and to complete the experience you'd really need a hydraulic chair.

Exploration games where you can look around and touch things (if you've got the glove) would be fun for a while. But what can you really do in these other that wander around, looking at and (to an extent) manipulating things?

In both cases the effect would appear to be there just for the sake of being there.

All I can say is that I'm interested, but what's it for?

Thanks to Division for the demonstration, but don't put me on the waiting list.

S.T.A.C.

The Simple Solution

Part (2)

Those Pilchards were great...make a back up working copy of your S.T.A.C disk, I managed to contaminate my working copy disk five times with various types of Virus.

Place the S.T.A.C disk in the drive and click on STAC.Prg when it has loaded you will be presented with the main menu press D then L this means Disk menu then Load. Put the cursor over QCK.ADV and double click, press ESC key. This has now loaded in the Quick start adventure file which contains a list of common verbs and some special functions that will make your adventure making easier to do. Remove the STAC disk and place a blank disk into your drive. Press D to go to the Disk menu then format your disk using this menu, press ESC key, press H to go to the High Condition menu then press insert, type in the following words and numbers, you must do this exactly as I have done.

```
High priority conditions

No.  Conditions

     if at 1 then 0 topcol 0 3 topcol 0 3 colour 0 0 colour 0 pause 50 goto 5 lf lf lf lf lf lf lf lf lf
 lf lf lf lf lf lf lf goto 6 3 colour 555 pause 200 3 colour 0 goto 8 lf lf lf lf lf lf lf lf lf lf lf l
f lf lf lf lf goto 6 3 colour 555 pause 150 3 colour 0 goto 9 lf lf lf lf lf lf lf lf lf lf lf lf lf lf
lf lf goto 6 3 colour 711 pause 200 3 colour 0 lf lf lf lf lf lf lf lf lf lf lf lf message 1 then 3 colo
ur 555 pause 5000 then 3 colour 0 lf lf lf lf lf lf lf lf lf lf lf lf lf goto 10 3 colour 555 3 topcol 5
55 0 topcol 0
```

This High condition has been taken dirctly from S.T.A.C itself, doing it this way should eliminate any publishing errors.

It is important that you understand the workings of the various routines so here is the High condition taken in small segments :- IF AT 1, simply means if you are at room 1 then do whatever command you have typed after this instruction, it could have been IF AT 488 or whatever but as I intend to start this adventure at room 1 then ergo the routine starts IF AT 1.

0 TOPCOL 0 3 TOPCOL 0 0 COLOUR 0 3 COLOUR 0, this line of command turns the TV screen black. The screen is split into

these four colour areas 3 TOPCOL and 3 COLOUR turn the text into black 3 TOPCOL is ther colour of the short bar description that runs along the top of your screen and 3 COLOUR is the colour of the text at the bottom of your screen. The other remaining colour commands are background colour for the top and bottom areas of the screen. the number after the command 3 COLOUR 0 is the RGB colour that you are changing the text to 000 is the ammount of Red 0 Green 0 and Blue 0 zero or 0 is the blackest colour. So if we had 3 COLOUR 700 then it would be Red 7 (most red) Green 0 (no green) Blue 0 (no blue) so the text at the bottom of our screen would be bright RED!! But black is what I need so it's 000 when S.T.A.C sees zero's then it automatically reduces them to just 0, don't know why but there it is. PAUSE 50 is Pause 50 = 1 second in time this is the formula to remember so pause 200 would pause on screen for 4 seconds...we need to pause the on screen action from time to time because if we didn't then S.T.A.C would make the say message go by so quickly that we would'nt have time to read it!! I am pausing the black screen so that people can 'see' the black screen.

GOTO 5 LF LF LF ....GOTO 6 3 COLOUR 555 PAUSE 200 3 COLOUR 0, this means that we will goto room 5 this room will contain some text then as we wish to see the text appear in the middle of the screen we must shunt it up using Line Feeds or LF or lf it will shunt the text up by one line per every line feed so sixteen line feeds puts the text into the centre we will now use GOTO 6 as an invisible type of brick to rest our line feeds on which will ensure that the text stays in the centre 3 COLOUR 555 PAUSE 200 3 COLOUR 0 this turns the text colour to grey (555) then keeps it on screen for 4 seconds. Then 3 COLOUR 0 turns the text black again which makes it invisible against our black background. Then we com...come t...we c...whats this.....sniff.....sniff.....Haddock!! I can smell Haddock.....have to go more next month.....One Eye.

Moving up to 68000 By Gerald T Kellett Part 2

Last month I started on this topic with an full article containing a proposed convention for register equivalents and op code analogs for Z8Ø programmers, here is some more.

```
 ADD r,n                ADD.B #n,r or ADDI #n,r
 SUB r,n                SUB.B #n,r or SUBI #n,r
 OR  r,n                OR.B #n,r or ORI #n,r
 AND r,n                AND.B #n,r or ANDI #n,r
 XOR r,n                EOR.B #n,r or EORI #n,r
  op r1,r2              op r2,r1 where op is any of the above.
 ADC  -
 SBC  -
 EX AF,AF'?             EXG.B r swap the lower 2 bytes of reg
 EXX?                   EXG.W r swap upper and lower words of reg.
```

EXG.B dØ would actually be equivalent to EX A,A' & could be used in place of the SWAP dØ,aØ suggested last time, similarly a series of EXG.W could be used for EXX instead of SWAPs if regs are being used for 16 bit usage.

My convention given last time did not give equivalence for seperate usage of the register pairs, it did suggest storing one single reg in the d reg the other in the a reg and use SWAP a?,d?. Storing them both in the lower word of any reg and using EXG.B can also be used, but some ops do not allow .B sizes and will corrupt the 'other half' when used, so the first suggestion is safer.

Using a combination of EXG.W and EXG.B will give you the equivalent of FOUR single byte regs in one 32 bit reg!

The 'address register indirect with fixed and data register offsets' ops are very useful, they allow you to move/copy/compare data without having to alter the addressing registers.

```
 another_lddr  move.b  Ø(a3,d1.w),Ø(a2,d1.w)   ;d1=1 less than number
               dbra    d1,another_lddr         ;of bytes to be moved
```

similarly:

```
 another_cpir  cmp.b   Ø(a3,d1.w),dØ       ;starts at end of table to be
               beq     found               ;searched
               dbra    d1,another_cpir     ;d1 as above
 not_found             ...............     ;falls through to here if
                       ...............     ;no match found

 found                 ................    ;d1 contains 'entry number'
```

This means you can perform repeated searches without having to reset the address register, I have used a3 which is the HL equivalent as that is the register used in Z8Ø cpir, equally you could use the IX & IY equivalents a4 & a5 which would be more in keeping with the 'fixed' addressing register role.

Strangely, since move does have these addressing modes, with cmp you can't compare register indirect to indirect or register to register indirect only register indirect to register or number, so you can't compare two blocks of memory directly, you have to load the bytes of one into a register and then compare. ie you can't have:

```
  cmp  (a2)+,(a3)+      or      cmp     dØ,(a3)+
```

you have to have

```
  move (a2)+,d0
  cmp  (a3)+,d0
```

The bit shifting ops of the 68000 are much the same as those of the Z80 (and the 6502 I think), it is only the mnemonics that are different from the Z80 ; ROR=RRC & ROL=RLC, SLA=ASL, SRA=ASR, SRL=LSR (there is offically no SLL in Z80 due to a 'bug' in the original chips, some chips have this correctly working, SLL=LSL).
Three things to note; you can specify by how many bits to shift (Z80 shifts by one bit at a time so you have to have repeated ops), and as well as shifting the bit shifted out into the carry flag it also goes to the extend flag (this is used if you wish to turn a signed byte into a signed word or a signed word into a signed longword), you must specify data size of .B for Z80 & 6502 equivalence.
There are no equivalents to RDL & RDR used for packed BCD, nor for RL or RR where the rotated bit goes via carry.
So in generalised form:

```
 shift_op.B     #n,r    ; #n number of bits to shift by
 shift_op.B     #n,(a?) ; -(a?) & (a?)+ etc also
```

Next issue...

I've some great stuff lined up for then, including some or all of these:-

* a Magnetic Scrolls interview!!

* a thought-provoking piece of fiction

* "Head's Cases III" - another dose!

* "AMOS Avenue" - the next column

* "STAC - The Simple Solution" Part 3!

* something about the "AGT" utility

* "Moving Up To 68000" Part 3!

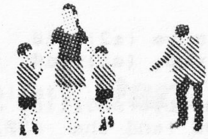
* "The Tales of Talespin" Part 3!

* an article on adventure standards

* hopefully much more as well!

I still need your continued help though, so please don't stop sending in articles and tips for your computer. Why not write to me and ask for a routine you just can't figure out yourself? That way I can write to one of my readers who use the same utility and computer as you, and they might be able to help! I can then publish the results in the magazine. Come on everyone - let's help each other!

LETTERS



Thankyou for Issue 1 of Adventure Workshop. There's quite a variety of articles, all except 1 were of great interest. The one I didn't like was the story on page 25. Rather silly, irritating and generally not my cup of tea! That's my personal opinion anyway. I'm quite interested in Talespin, I don't have the program, but I'll be interested to see how Brian writes his games.

Two STAC columns, eh? By the looks of them, they'll complement each other nicely. Ian Eveleigh's article on page 8 makes some very valid points. People who put down utilities without seeing them are just ignorant. It would be like me telling you that the carpet in your living room doesn't suit the curtains. Seeing as I've never been remotely near Baildon makes my argument pretty pathetic!

Ian's AGT review was very interesting as I was thinking of getting a PC emulator for the Amiga. It told me what I wanted to know, but I'll be reading with interest. I could even do one on the PD ADVSYS for the Amiga. ADVSYS actually has quite a good way of going about writing an adventure, except it's a little difficult...

The machine code article was completely unintelligble to me (that means it's good!) and the CES report was quite an eye-opener.

Altogether I really enjoyed the first issue. I like the spread of articles. Can't wait for the next issue!

John Ferris, Coventry.

AGT sounds quite good, I've not seen it on any of the Public Domain library lists I have, how new is it? With still no sign of ST-PAW I was about to get ADVSYS instead, but I won't now.

Since it compiles from WP files, and ST disks are fairly interchangeable with (3.5", 5.25" if you have a 2nd drive of that type) PC disks, the ST can read 360/720K SS/DS PC disks, and PCs can read ST disks likewise providing you use a formattor that puts the 'media descriptor byte' on the disk. Then an adventure can be written on one or the other and compiled on both, providing you have access to both type of machine.

So Ian Eveleigh, you've got the PC version, I'm going to get the ST version, but is your PC fitted with a 3.5" or 5.25" disk drive?

(By the way the book 'Computer Adventures - The Secret Art' can now be had at £4.95 by anyone according to the ad on the back page of Adventure Workshop issue 1, so is there still a discount if you buy AGT?)

On the subject of AMOS, I would like to ask John Ferris if it is fully compatible with STOS (at least as far as the non-graphic parts are concerned)?

It's just a pity STs and Amigas can't read each others disks, since even if the '.BAS' files are not encoded that same '.ASC' listings could be 'ported' from one to the other. Unless anyone knows of any programs that allow STs to read Amiga disks (the ST can cope with disks having 11 sectors per track as Amiga disks have, or is it just that the directory is 'wrong') or vice-versa?

Gerald Kellett, Stamford.

I hope you are going to feature some HATRACK II articles in the future in Adventure Workshop. It's the best adventure creator for the Amiga (especially bearing in mind the competition) and even though the PAW is promised by Gilsoft in the future - that could be a few years away yet!

Moving on... I enjoyed reading all of 'AW' especially the Talespin review-ette, though the more technical sections, like the Moving Up To 68000 section, was too overwhelming - but I suppose you must cater for those technically minded boffins too!

Tim Kemp, Norwich.

incentive
SOFTWARE

Zephyr One, Calleva Park,
Aldermaston, Berkshire,
England RG7 4QW.
Telephone 0734 817288.
[Facsimile 0734 816940]

Mr C Hester
3 West Lane
Baildon
Near Shipley
West Yorkshire
BD17 5HD

11th March 1991

Dear Chris

Thanks for sending the first copy of "Adventure Workshop" - it's an interesting and informative magazine and deserves to do well.

I'd be glad to help with queries about STAC, but not too many please! I'm already up to my ears in work. Genuine questions are okay, but please don't ask me to rewrite the manual!

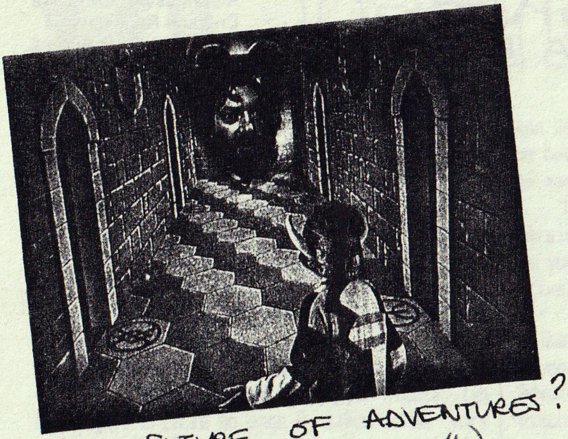
A couple of things - on page 15, Gerald Kellett mentions that we are giving away the STAC source code. THIS IS NOT TRUE. The whole rumour started from one magazine (I can't remember which one) which printed this about 9 months ago. What we are releasing is the source code for you to add your own extensions to STAC, which is a 3 page document. I am enclosing this, which you may wish to print in the magazine.

I liked the detective story, but it took me about half an hour to translate it into English!

Anyway, hope it goes well.

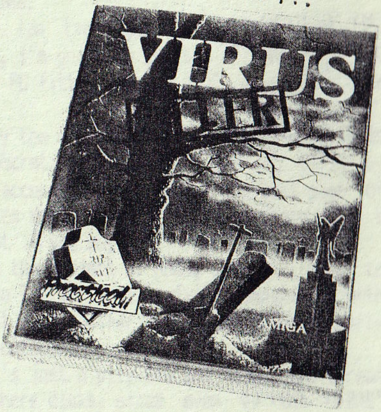
Yours Sincerely

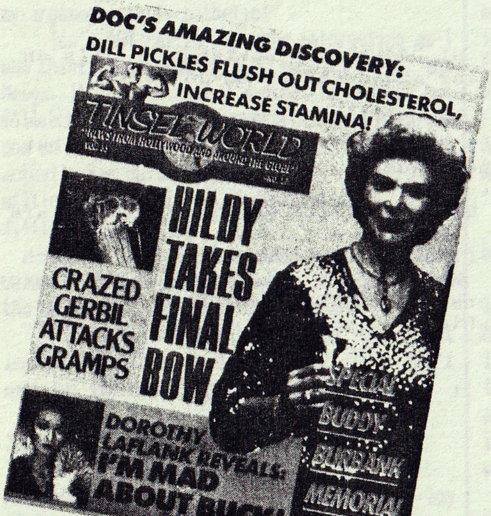
SEAN ELLIS
AUTHOR - STAC



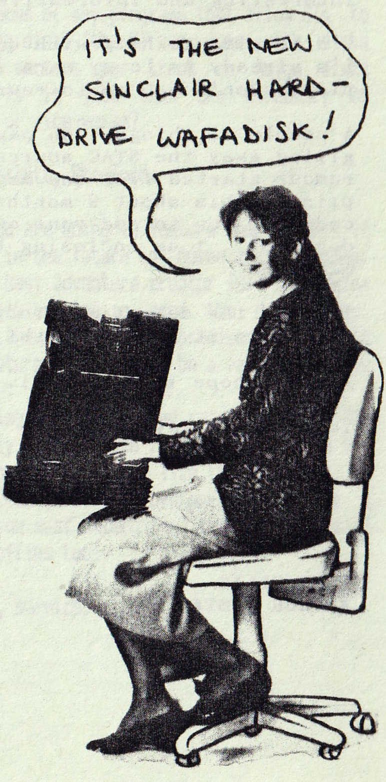
THE FUTURE OF ADVENTURES?
(ITV'S "KNIGHTMARE".)

THIS SHOULDN'T BE NECESSARY...




DOC'S AMAZING DISCOVERY:
DILL PICKLES FLUSH OUT CHOLESTEROL, INCREASE STAMINA!
TINSEL WORLD
HILDY TAKES FINAL BOW
CRAZED GERBIL ATTACKS GRAMPS
DOROTHY LAFLANK REVEALS: I'M MAD ABOUT BUCK!
SPECIAL BUDDY BURBANK MEMORIAL ISSUE

THIS IS PART OF INFOCOM'S PACKAGING FOR THEIR GAME "HOLLYWOOD HI-JINX". PIRATES DON'T KNOW WHAT THEY'RE MISSING!


IT'S THE NEW SINCLAIR HARD-DRIVE WAFADISK!