

DID YOU KNOW?

*We still have the very best PD library
for the Atari Classic*

There are over 400 disks available

Many disks are only £1.50 each

Collections and special sets are even cheaper

**EVERY PURCHASE OF A PD DISK
HELPS TO SUPPORT THE MAGAZINE**

DID YOU REALISE?

We still have the entire ST library available

There are over 1,000 disks

Disks are only £1 each

We will send you details on request

**KEEP SUPPORTING US
AND WE WILL KEEP SUPPORTING YOU**

Page 8 Publishing's

NEW

ATARI USER

The Resource for the ATARI CLASSIC and the ATARI ST

Issue 84 - May/June 1998

£2.50

FOR THE ATARI CLASSIC



- ❶ **THE SEARCH IS ON**
Search and sort routines
- ❷ **USEFUL USR**
Crack the connection between
Basic and machine code
- ❸ **CD COLLECTION DATABASE**
The next best thing to a CD cabinet!

THE WIDER SCENE

**FINDING A
HOME ON
THE INTERNET**



PLUS ... WORM W PARADISE ... HAPPY TYPER ... SPARTANOS ... MAILDAG ... AND MORE!

For one of the few times in the past ten years or so I can't really think of something to write me about in the editorial so I guess I'll go to the old standby of asking for your contributions. In fact this isn't really a standby but an important topic at the moment for if you don't contribute something we will soon be struggling to fill the magazine. We're okay for the next couple of issues but there is very little to reserve and I need to think far ahead. Whenever I make these pleas for contributions the old faithfuls such as John Foden's always respond with another disk or two full of articles and programs and, whilst there are more than appreciated, we can't keep relying on the same contributors time and time again. Apart from anything else these people will probably dry up one day or don't rely on others to make the contributions, try and think of something you can do. Of course not everyone can write programs or compose articles but anyone can write a letter and that is a good starting point. A full mailing is a solid foundation for any issue and you can begin by responding to any topic that interests you in this issue's mailing. If there is nothing much of interest then write to us about something you are interested in, it doesn't even have to be about *Just* related computing, as the recent interest in the internet has shown.

If you want to do more than just write a letter, then a good idea is to take your favourite program in PD program is probably best since these are readily available and do a comprehensive write up on it. If you use a program regularly then you should know it inside out by now. Share that knowledge with others. Don't be afraid to do an overview on similar types of programs, for example comparing the various word processors that are available. How about taking a PD program and writing about what isn't in it? Well readers about what you would like to see in your dream program and maybe someone will be inspired to write it.

So what other ideas are there?

- Or send in a favourite programming routine you use often and tell us how it works.
- Polish off a program you have written and send it in.
- Find good programs from other publications, as Dennis Pegearty did this time.
- Find an article somewhere that you think could be written better and make it interesting for us all.
- Dig out some good PD programs we don't have and send them in to be used as disk banners.

And there is only a few of the ways you can help!

THIS IS IMPORTANT

I have said many times that this is your magazine, full of your contributions and you must stay committed for us to survive. With the pressure of work in trying to keep the magazine paid I simply don't have the time to fill so huge volumes of the magazine as I may have done several years ago so I am relying on you more than ever to make my job a lot easier, and to make *New About User* as interesting as it has always been.

You now it makes sense. Please start thinking now and writing tomorrow! Once you start, it becomes a lot easier and when you see your name in print you will be inspired to contribute again. That's our future.

Les Elingham

3D RIPPLE

Raphael Espino presents this neat little program for a great graphic design

This program creates a 3D ripple surface which serves no purpose except that it is interesting to look at since it is complete. It does take a while to run so you will need some patience but the result is well worth it. I suggest that you run the program with Turbo Basic which will speed up the drawing to some

try compiling it with the Turbo compiler. Remember to save a Basic version though as you might want to experiment with the parameters.

MODIFYING THE PROGRAM

If you want to try and modify the program note that the main variables are:

- WSR width of the surface
- WSH height of the "bump"
- DSM apparent depth of view
- ANG0 angle of view (in degrees)
- PS01 height of screen in pixels
- PS02 width of screen in pixels
- CSX centre of shape horizontally
- CSY centre of shape vertically

To use a different graphics mode change the GRAPHICS command in line 105, and variables PS01, PS02, CSX and CSY to suit your new mode. For example, mode 7x14 would be PS01=80 PS02=160 CSX=80 CSY=80. CSY should be roughly one quarter of PS01 and CSX half of PS02. You should also change the values in the STEP commands in line 20 by dividing them each time the screen size is halved.

To change the shape's position on the screen then change CSX and CSY and to change its height and width then change PS01 and PS02.

Try also changing the following lines:
 Line 89 - increase the number in the STEP commands to speed up the drawing. This will make the drawing "blockier" however.
 Line 49 - the second operand (PS02/4) is the one that determines the shape of the surface. Also try (PS01/2)/4 and (PS01/PS02)/4 for different shapes.

```

10 1 RND
21 2 RND : 3D RIPPLE BY RAPHAEL ESPINO
31 3 RND
40 10 GRAPHICS 24:COLR 1:CH 148:SH 24:V
74:DSM=20:DSM0.14:SM 100000:RND :5
74:CSX=80:CSY=80:RND
51 20 PSW=10:PSW=128:CSX=148:CSY=80:CSZ=70
74:TRAP 148:SH=324:PSW=34:CH=10:VSH
60 30 PSW 30:SH 10 : STEP 1:PSW 1*25 70
25 STEP 4
70 40 V=220:RND*100:V/10:SH=128:V/10:V*10:
57:CH=34
80 50 11 1*25 TRN 180:V*10:V*10
90 60 11 1*100 TRN 180:V*10:V*10 CH=10
4:PSW=21-(1*100/20):PLUT (PSW-1)*10+
1:PSW-(PSW-1)-(1*100/20)
70 70 11 1*100 TRN 180:V*10:V*10 CH=10
4:PSW=21-(1*100/20):PLUT (PSW-1)*10+
1:PSW-(PSW-1)-(1*100/20)
21 80 NEXT I:END X:END 77,120
30 90 GOTO 10
40 100 TRAP 100:GOTO 10
    
```

Mailbag



This issue's
Mailbag
conducted by
Les Ellingham

INTERNET SUCCESS

After describing my experiences on the Internet last issue, Alan Pedlow came to my rescue and provided a couple of pages of Web addresses relating to Leonard Fellor and other 'hacker' of internet. He obviously has the experience to find what I could not. Some of his comments will be of general interest so here are a few things he had to say: "Having dabbled with the Internet and creating a simple home page when I first acquired a PC back in 1986, I've now a well-used ephemeris for Iceland that is cybernetic. Part of my spare time is spent in maintaining the Web page relating to The Crewlers - Basingstoke's award-winning youth scouting club which my daughters are involved, and also to maintaining the Web

pages of the British Youth Board Association (BYBA) - the national organisation which co-ordinates and promotes youth learning. In case anyone's interested, the URLs for these pages are <http://www.the-crewlers.org.uk/> and <http://www.byba.org.uk/> respectively.

Your editorial recounted your attempts to research Leonard Fellor on the Internet and you were surprised that using the Yahoo search engine you only retrieved a handful of hits. This may be down to a couple of things: firstly, the content of the list continues to grow exponentially, keeping track of it is more than a full time job; secondly, web pages are indexed by tags within the HTML code which list relevant keywords for search engines to use. If a word or phrase you're interested in isn't included in the tag, it won't be indexed. There are some search engines which build their indexes by scanning the web and the actual content of pages, but of course they've got to find all the pages.

I did a quick search via Webcrawler and came up with a number of hits concerning Leonard Fellor and enclosed a print out of various addresses which may be of use next time you go cybernetic. As opposed to a

Cyber Cafe, have you tried your local library? Many provide internet access, I understand, of some form." Alan's list of addresses was superb and has enabled me to get much more value from my second floor. As to local libraries, some people are lucky, others not, depending on the local authority. Southampton libraries have no public internet access of any kind as they "are installing a policy decision". I do know that the main library in Bournemouth offers free internet access, all you have to do is book your library admission. Also the library in the little village of Street in Somerset offers public access, although I don't know if you have to pay. If they can do it why can't all libraries?

It would be an excellent idea if we could put together an article on public internet access with the help of our readers. If any of you know of Cyber Cafes in your area, find out what they charge and send us the details together with their postal address. If you use the Internet at your local library let us know details of what they charge and what their policy is, for instance do you have to be a member of that particular library or is access open to anybody? Is there such a thing as 'libraries on-line' on

the Internet that will give details of all the libraries that are connected up? There's a good chance for some of you to contribute to an article, even if you can't write the article yourself (although a volunteer would be more than welcome). The Internet really is a fabulous tool for research and it would be good if those of us without PCs or modems could get to use it a little more easily.

TRANSFER REQUEST

Dennis Hodges from Southampton has been having a few problems transferring data between the Atari and his PC but has worked out some of the problems as he explains: "With regard to Philip Brown's problem with the G.M. Transfer cable, I had the same trouble of no interaction between the PC and Atari. Having printed out the 80 pages of information from the file <http://www.com.on.ca/~steak>, and anything else I could find to read, I must say that Nick Kennedy really does explain everything very well, although I did not understand a lot of the technical parts. Having read it all, I still could not get the Atari to recognise my PC. Anyway I did

manage to crack it then. Get MODEPC up and running on the PC, then load MYDOS453.KIB on to the virtual Drive C. Then, press A on the PC keyboard to get the starting screen up. Press T which toggles the status of command line input. Switch on the Atari and the command line at the bottom of the program should change to DRV 31 at 4F. Type DOS on the Atari and you should get the DOS program on the monitor. I did experiment with some of the settings, but this did not seem to make much difference, only on UART.

There seems to be a lot more to transferring games and files than it made out but I hope that this will help someone."

I thank Dennis. I have to admit that I didn't understand much of it, but I guess if you've got the program up on the monitor in front of you it will make sense. I hope that Philip is sure up and running and maybe some others will be helped as well.

MORE INTERNET

James Austin's letter written just a couple of days after I had completed the last issue so refers to items raised in

Issue 62. It is still interesting as it's four years down: "First of all, regarding the letter from H.S. Wood, could he, or anyone else, explain what a P.L.C. is? Presumably they are something to do with PCs? As regards connecting an Atari to a PC, I don't currently own a PC and don't intend to get one in the near future, but I do find it interesting to read about, as long as the subject matter does not get too technical, because it then goes way over the head of users such as myself who do not own a PC. It's okay as long as it doesn't take up too much of an issue and so overshadow other topics. It hasn't happened yet but I feel that we should be careful not to let this happen."

On a similar note, I have found reading about the Internet, in John S. Davidson's *Joeyway Into Cyberspace* very interesting. It is a pity that it is so expensive though. I have only been on the Internet once or twice at school, when the teachers weren't looking or when it has accidentally been left turned on, which isn't often, but have managed to find several Atari-specific Web sites, including an ATARI 6-bit web site at the University of Kent, at Canterbury (presumably there are similar web sites at other universities?), also a





18

"Digital Audio/Digital Imaging" web site where all the articles/programs from the magazines can be found and downloaded. There is also a really interesting site where screen shots and information on all the classic Atari games can be downloaded and viewed. There also appears to be a lot of information on the Net about new ADAM commercial games from the States. I never had the time to write down any of the Web site addresses for these but I found them pretty easily using the HotImage search facility, so perhaps John or someone else with access can help a look around for some of these sites and maybe send in information about them, maybe also writing in any downloaded articles/information that might be of interest to readers. Perhaps a regular section could be set up in the magazine. In addition to John's Denton's columns, where this stuff could be used in and published for everyone who can't get Internet access.

Before I close, a couple of notes for help. Firstly, does anyone out there own a 1600? If you do then would anyone care to test a new piece of software that I have written for it? If so PLEASE contact me at the address below. It will come with a

manual fully explaining how to use it and detailing all of its features. I'm asking this because I really need someone impartial to test it out for me to see if it's worth me completing it and maybe sending it in to PAGE 6. Also, does anyone know where I can get a full working version of Daisy Disk II for the one that allows you to use multiple fonts in a document, has SpartaDOS II support, etc? If you do then please write to me at the address given."

■ Thanks for some interesting points James. Your comments about using the Internet at school seem to echo other school's comments of the Internet. At my son's school they got connected up several months ago via a special deal with the local cable company which effectively cost them peanuts. When they first talked about it they said that they could probably offer you and internet access at around \$60 a hour, and actually make a profit from it. It has turned out however that we have been severely restricted so that pupils can only use the internet during school time in connection with school projects only in the presence of teachers. What a total waste of such an important resource tool! There are many responsible children (my son included) who would love to use the In-

ternet in their own time, especially at low cost, who are being denied such use because the school can't work out how to use and supervise the system properly. They are basically scared stiff that people will use the Internet to access pornographic material and because they don't know how to control this they ban all pupils. Not only is that copying out from teaching a problem, it is also an insult to the majority of responsible children who could expand their education enormously in their own time through accessing information on subjects that they are genuinely interested in. Maybe one day it will get sorted, but I must confess that I have little regard for the majority of teachers in State schools, who seem to be interested in little beyond their immediate responsibilities.

As regards James's requests you can write to James at either at 18, Clive Road, Green Park Estate, Buntingford, SG 10 6JH or to me at 18170 JLU.

BIND IT

Rolfe Jones Pass a couple of contact addresses and an idea for a binder you can make to hold your copies of PAGE

"The following telephone numbers may be of use to owners of EPSON equipment - EPSON 0800 209022, department required is by number selection. If you require spares or optional add on parts they will refer you to their distributors. Try also Mike Pastores (Munday) on 0800 209020. They are a Mail order company and their prices are cheaper than any of the Dixon Group.

If you want to have a go at making your own binders for MAU, purchase an M10000 BUNDED Binder No 10420 for approximately £1. It is a perfect size and well backed and it will hold 6 issues. Also buy a pack of ELASTIC COORD 1/4"Thin-Strap, about 5 metres long. It will cost you about another £1 from Hobbiesupply Shops, Southampton, Tesco etc.

Remove the metal spring clip using a pair of pliers that will grip the lower end of the rivet and allow you to push it out. The elastic is cut into approximately 13cm lengths but you may like to experiment with the length, so you can get the tension you prefer. At the top of the spine on the outside make a mark 15cm from the top and stretch the elastic into seven spaces giving 6 marks where the 6 pieces of elastic will be glued. Use 15cm of the elas-

tic unstretched to glue here and then do the remainder. At the bottom end of the spine repeat on the top, once the top glue has set. You will find the binder will bend, but when the magazines are fitted all will be well. The proposed glued ends can be covered. I used sticky tape, black at the bottom and red at the top.

If you cannot get the soft backed type of binder then use the hard backed variety. These are used approximately 4.5mm more. Somebody may have a better idea, if so let's hear it."

OTHER COMPUTERS

Regular contributor Knute Cooker has quite a number of things to talk about. I have just received the latest issue of N.A.U. and must say that I was most impressed. As usual, it was full of interesting articles for people of all levels of knowledge, making an excellent read.

I'm now attending university at Luton and as such, I now have an e-mail address by which other Atari users can contact me: KJ-COOKIE@WY0.LUTON.AC.UK. There is only so much help I can give people as I don't have my Atari

with me in Luton (for reasons outlined in the rest of my article) but I will do my best to answer any questions relating to general Atari use and am quite willing to have a chat with anyone. Please add my e-mail address to the MAU Internet contact list.

My time at university also gives me free access to the Internet, so I was interested to read of your own experiments with the World Wide Web (WWW). I have tried to use the WWW in the past to do some serious research for university projects, usually with limited success. For example, recently I have been researching into a phenomenon known as "Think Building Syndrome" (TBS-B). However, when typing "TBS" into the search engine, I received information on loads of subjects not in any way related to this phenomenon (such as the Special Boat Service and something about counting). The stuff that WWW relevant to the subject was mostly written by companies trying to sell items to alleviate the problem.

Another danger with the Internet is taking everything that you read as it is face value. One of the biggest problems with the Internet is that anyone can create web-pages with biased, prejudiced views and present it as fact.





A lot of the so-called "hate" writers by "reports" are extremely hard to verify, with pages disappearing from the Net as quickly as they appeared. It's also surprising at how many people use the Internet, considering how long it takes to load the pages you are looking for. It's little surprise that it's been dubbed "The World Wide Wait!"

In these 800's mailing, Joel Goodlin suggested that other readers may like to hear of Atari enthusiasts' experiences on other computers. Recently, I've been using PCs and an Amiga rather a lot. Here's why.

I've recently sold my Atari ST and used the money to buy an Amiga. OK, I hear lots of outrage and surprise from the Atari community, but let me explain myself. I bought the ST as a step up from the 8-bit for disk-top publishing and, although it satisfied me at first, I soon became annoyed at how the program I was using kept crashing, partly due to bugs in the software but mainly due to my machine only having 1 Mbyte and no hard drive. The cure was simple - buy more memory and a hard drive, at what cost? It turned out that if I sold my ST and software for \$700, I could buy an Amiga 1200 with hard drive, 4 megs of memory, an extra floppy

drive, and loads of software for only \$400 more. If you look at the prices that Atari units generally wind for (money and/or systems with hard drives aren't hard), you'll be looking at a lot more money than one hundred pounds for at least you would have been seven months ago when I bought the Amiga). At the end of the day, money had to talk, and with the collapse of Atari, I saw little loyalty left to a company who had shown me none. And, as much as I liked the Amiga, I have to admit that the Amiga is FAR superior. The Amiga's strengths are not limited to 8 characters (in fact they can be about 20 characters long). The desktop is easier to use, especially with the various shareware improvements available, and I can't remember any software ever having crashed (unlike the ST, where even half-meg games would crash for no apparent reason). The joystick/mouse ports are in a much easier to access position, the keyboard feels nice (I never could get used to the ST keyboard) and the various software is cheaper and will bring support/updates. So in my own little world with Atari over?

No! My computer may seem unworldwide and it's true, I don't have my 8-bit at university with me (because of space limitations and the fact that my Amiga can save less files straight onto a 3.5" disk and load them into a PC, which is essential for my university course) but you just can't beat the old 8-bit for easy to use, easy to adapt (and I have a feeling I may buy an ST again when I have more money but for now, I'll be using my Amiga and Atari 8-bit with satisfaction.

It's interesting to hear people criticize the 8-bit and 16-bit computers and rave about their powerful PCs, only to later prove that they don't even know how to use a fraction of the hardware available on their PCs. For example, a student in my class made a big deal of the new Pentium computer she had bought when she started the degree, yet for six months when she thought she was saving her work onto a blank disk, she found she had been saving it to the hard drive! Therefore, all the work she thought she had saved to one floppy disk had actually been spread amongst the numerous computers she had used at university, and the one she had used at home!

In another instance, when I was doing a group project for one of my degree modules, I gave one member of the group my work on a disk, along with a printed copy, so

she could paste it together with each group member's work during the weekend to create a finished essay. When the girl got home and tried to load it into her Pentium, she was shocked to find that there didn't appear to be any file on the disk. After checking, she ended up typing in all of the work from the printed copy (about 5 pages). After the weekend, she complained to me that the disk was empty. It turned out that her word processor was only looking for files with an extension of .TXT and because my file was saved with a .WPP extension, it hadn't shown up on the directory! All she had to do was change the type of file that would be listed in the directory from ".TXT to *", which she didn't know how to do!

I'm also amazed at the number of PC users who can't even copy or format a disk! And would you believe that you can't swap floppy disks while using programs in Windows for "Win-95" as it's been called because it corrupts the data table on the disks, effectively losing everything on the disk!

I think that these examples go to show that it's not just the capability of the computer that is important, but also the capability of the user!

† Interesting coincidence,

she could paste it together with each group member's work during the weekend to create a finished essay. When the girl got home and tried to load it into her Pentium, she was shocked to find that there didn't appear to be any file on the disk. After checking, she ended up typing in all of the work from the printed copy (about 5 pages). After the weekend, she complained to me that the disk was empty. It turned out that her word processor was only looking for files with an extension of .TXT and because my file was saved with a .WPP extension, it hadn't shown up on the directory! All she had to do was change the type of file that would be listed in the directory from ".TXT to *", which she didn't know how to do!

PROGRAMMING BASIC LISTING



Kevin, which I think shows the difference between the computer enthusiast and the computer user. Those of us who came into computing via the Atari generally have a greater understanding about how things work and so will benefit from using any kind of computer. The vast majority of PC users don't understand the slightest when it goes on which is why you get all these "amateurs" typing people off with training software and the like. It is unfortunate that the amount of money that schools and businesses waste because the people working there can't read a manual. It also annoys me that people need to go on training courses to learn how to use a computer. I don't mean how to write programs, but just so that they can learn how to press keys! How many of you reading this had to go on a course to learn to use the Atari?

I'm also amazed at the number of PC users who can't even copy or format a disk! And would you believe that you can't swap floppy disks while using programs in Windows for "Win-95" as it's been called because it corrupts the data table on the disks, effectively losing everything on the disk!

I think that these examples go to show that it's not just the capability of the computer that is important, but also the capability of the user!

† Interesting coincidence,

Kevin, which I think shows the difference between the computer enthusiast and the computer user. Those of us who came into computing via the Atari generally have a greater understanding about how things work and so will benefit from using any kind of computer. The vast majority of PC users don't understand the slightest when it goes on which is why you get all these "amateurs" typing people off with training software and the like. It is unfortunate that the amount of money that schools and businesses waste because the people working there can't read a manual. It also annoys me that people need to go on training courses to learn how to use a computer. I don't mean how to write programs, but just so that they can learn how to press keys! How many of you reading this had to go on a course to learn to use the Atari?

NEVER TOO LATE

Paul Herbert starts off by letting us know that it is never too late to start writing to Mailbox. After nine years of reading Page 8/25A11, the finally gotten off my backside and written to!

First of all, some Translist problems. Paul and Brad Orr (like both seem to lead in one chunk, making it impossible to transfer in stages, and the programs are too big to hold in my CD's 84k capacity (with Translist loaded). Shame, because they are two of my favourites, especially H/H. Red Max, on the other hand, loads in enough stages to successfully transfer to disk (514 sectors Paul), but doesn't seem to load from disk. It's not my copy of the game, because I loaded it from tape with success (had an hour later.... those were the days), so something went wrong in the transfer. Is there anyone who has Translist-oh of these games successfully who can help?

Raking through my old tape collection I came across three cassette that I had long forgotten about, and I would be curious to know if anyone else has any copies of them. They are issues 3, 4 and 5 of a cassette magazine called Atari Computing, dated 1984. They have various games, utilities, and reviews (including a review of Red Max when it first came out) on them, plus a three part text adventure (which incidentally in Basic) called the Keys of Time. It's a help mag any folks. The address was 1



Golden Square, London W1R 6AB (Argus Press, Solihull). Does anyone know what happened in Argus Press Solihull? If the copyright status permits it says (AMPS on the tape) I would be more than willing to transfer these onto disk and to send them in for Domes (Argus or Public Domain - we all know how about us these days).

On the subject of potential Domes Programs, I have a book of type-in games called Games For Your Atari (Frigate, 1982), many of which I have typed in. Needless to say, some of them are pretty awful ("Compliment Generator" - random phrases to boost your ego), but there are some real gems in games lurking around, and again, copyright permitting, I would be happy to send them in and share them with other users if possible.

Finally, I discovered the real cost of dimes. Recently, a letter came toward my house and required as to the nature of my computer, given that it clearly wasn't a PC - surely they are the only types of computer in the world? Wrong again. After telling him that it was an Atari and had having revealed the usual stack-up nature that I have come to expect, I asked him for a few minutes of his time, and loaded up the "Dis-

ks Domes II" disk (I had it in the Page 8 PD library if anyone's interested). This has a collection of dimes, but I went straight to the one with the notation of a dime not owing to the dime (one from Domescom AMD) (one simultaneous word search happens 490K compressed onto a standard 120K disk). After he ate his words (voluntarily), I informed him that the processing technology was older than me (being a fledgling 20 years - young for a human, but in computer terms that's about 40 generations). Doesn't they don't make computers like they used to?

↑ I don't think anyone ever read Atari Computing. Post input from you because we had a full colour ad in the back of the card which accompanied issue 1 and we didn't receive one single response! After much complaining we actually got one cheque for £200 book, ordered in at advertising circles. In fact it was an unusual for anyone to request advertising money that I kept the cheque as a souvenir. It is dated 2nd November 1984 and is kept in a little tin along with another cheque that someone once sent us for postage, made out for just 20 pence! I didn't bother to cash it!

**WE NEED
YOUR
LETTERS
ARTICLES
PROGRAMS
URGENTLY
PLEASE
CONTRIBUTE
NOW**

(It's your mag after all)

Another decent load of letters this issue for which many thanks. Keep your letters coming, folks, they really do mean a lot. As always the address to write to is:

**MAILING
NEW ATARI USER
P.O. BOX 54
STAFFORD
ST16 1DR**

XL/XE PROGRAMMING

BASIC LISTING SEARCHER

*John Foskett
shares one of the
utilities that help
him write so many
great programs*

USING THE PROGRAM

Firstly, the input file - the BASIC listing to be searched - must be saved to disk as an ASCII file, that is to the LIST format.

When the BASIC LISTING SEARCHER is run, a fixed screen is presented with a prompt to enter the file name of the input file. The default file name of PROGRAMLIST can be entered by simply pressing RETURN and it is convenient to LIST the input file to disk using the default file name.

The next step is to enter the string of characters at the next prompt using a maximum of 12 characters. After entering the string of characters for the search, either RETURN is pressed to continue or ESCAPE is pressed to exit. Upon pressing RETURN the input file will be read from the disk and all lines which contain a match with the character string will be printed on screen.

All lines which contain at least one match with the character string will be printed on screen with all the inverse characters they may contain being converted to normal. All matches with the character string are highlighted in the lines in upper case inverse video for clarity. Note that the input file is not changed in any way by the program, the lines are only printed on screen in this way for clarity.

The search may be passed at any time by

When writing or modifying a BASIC program, it is sometimes a necessity to find the number of occurrences of certain things within the listing. As the listing grows, however, the effort needed to search the listing also grows and it can become very time consuming, error prone and very tedious. Sometimes a line number is accidentally typed wrongly, such as entering line 1005 instead of 1008 and as a result, the line suddenly matches into this air! It is essential that such errors are corrected immediately while they are fresh to the programmer's mind otherwise they could become forgotten, and maybe even result in bugs. It is from this need that the BASIC LISTING SEARCHER was born.

The BASIC LISTING SEARCHER is a utility program that will find every occurrence of a specified string of characters no matter where they are to be found within the listing.

pressing the ESCAPE-BAK after which RETURN may be pressed to continue or ESCAPE may be pressed to exit.

At the end of the search, the number of times the character string appeared in the listing is displayed on screen together with the number of lines that the character string appeared on. At this point ESCAPE may be pressed to exit or RETURN may be pressed to list the line numbers upon which the character string appeared after which ESCAPE may be pressed to exit.

TECHNICAL DETAILS

ERROR TRAP

The error trap routine is responsible for detecting the End of File (EOF) error 136 and returning control back to the main program. The error trap routine is limited to only the errors associated with reading a disk file.

DISPLAY LISTS

The program uses two display lists, a normal mode one, but final screen is used for the main display and a special four-line mode one screen is used for displaying disk errors. The display lists are defined together as `DIS` and `MOVDIS` into page 6 at address 1596. The address of the main screen display list is 1594 and the address of the error trap display list is 1590.

THE CURSOR

The program uses player one as the cursor for keyboard entry and the VBI routine is used to flash its colour between the two preset values (144 and 180). `STP` loaded with `space` (the heart character) is used to clear

the player strip by `MOVING` 256 bytes of its length into it prior to defining the cursor shape again using `MOVE`.

PRINTING A LINE ON SCREEN

Prior to printing a line from the input file on screen, the display flag at location 794 is set (`POWLD` with 1) to enable the ESCAPE-CONTROL characters to be printed on screen without the computer acting upon them. After printing the line, location 798 is reset to zero.

FINDING A MATCH

Over a line from the input file has been read into `LAR`, the program uses `LEASTN` and `INSTR` to find all matches with the character string. The first `LEASTN` command found on line 240 attempts to find the first match with the character string and therefore determines whether or not a line is to be printed on screen. Once the first `LEASTN` command has found a match, the second `LEASTN` command on line 250 attempts to find more matches with the character string.

Each time a match is found, the line number is stored in `STP` for listing if required after the search. The line numbers are stored using the normal 4 byte `byte` method.

VBI ROUTINE

A small deferred VBI routine defined as a string (`VDM`) is used to enable all lower case and inverse characters to ease keyboard entry. Because `USEDSB` is used to find all matches, the status of any letters in the character string is unimportant since `LEASTN` will find all matches irrespective of case or mode. The routine is also responsible for flashing the colour of the `PMO` name between two preset values (144 and 180) for disabling the attract mode and for disabling the `COMBOL-1` strip start logs.

PROGRAM BREAKDOWN

To help study the listing, a breakdown of the program procedures, labels, strings and major variables follow...

PROCEDURES

BEEP	The beep
CLICK	The keyboard click
CURSOR	Controls the horizontal position of the cursor. Used in the <code>INPUT</code> procedure
INT	Initialising routine
INPUT	For entering data from the keyboard
IPSUB	For printing data on screen. Used in the <code>INPUT</code> procedure
SPACES	Removes any leading and lagging spaces from it
WAIT	Pauses the search

THE ONE LABEL NAME

a REPRN Resets the VBI vector and reuses the program

STRINGS

CH#	The character string entered from the keyboard converted into inverse characters
CH1#	The character string copied from <code>CH#</code> before converting into inverse. As entered from the keyboard
CUR#	For defining the cursor
PS	The file name copied from <code>FR#</code> but with 'D'
FR#	The file name entered from the keyboard without 'D'

IS	Stores data entered from the keyboard used in the <code>INPUT</code> procedure
LS	Defined as a line
LNO	Stores the lines of BASIC read from the input file
NO	Line number of the current line as a string
ST#	Used for loading the display lines into page 6, for clearing the <code>PMO</code> strip and for storing the line numbers of the lines which match the character string
TS	The program title
VBI#	The VBI routine

MAJOR VARIABLES

CUR , MAX , POS , PRV , X# and YY	General variables used in the <code>INPUT</code> procedure
FM# (from <code>LEASTN</code>)	Equals zero if no match is found and sets the loop. If greater than zero, a first match has been found
LEN#	Length of <code>CH#</code>
LINES#	Number of lines that a match has been found on
LNUM#	Line number of the current line
MATCH#	Number of matches found which includes more than one per line
PM#	Address of <code>PMO</code>
ST	Pointer for the next position in <code>ST#</code> to store the next line number upon which a match has been found

THE LISTING

The full listing can be found on this issue's disk. If you prefer to type in the listing a TOPS coded/printed listing is available on request, see inside back cover for details.



ETS

JUMP takes you to the nearest hub
SAY HOME takes you home
 and I think that
SAY EXIT takes you to the SCOT.

I found Lee Williams' system for ETS somewhat confusing and inconsistent, but I am probably biased towards my own system. To use it, translate the current tile into a number, and the location tile into a number. If the location number is larger, move 'W'. If it is smaller, move 'E'. Always ignore the first one out to the address, and aim to make up the difference by entering in the ring with the nearest repeated three number in the difference. Examples are as follows:

White	1	Walkway	1
Grey	2	Outer #1	2
Violet	3	Outer #2	3
Blue	4	ET#1	27
Green	5	ET#2	85
Yellow	6	Central #1	343
Orange	7	Central #2	729
Red	8	ET#1	6501
Brown	9	ET#2	18663
Black 10/0		Inner #1	89049
		Inner #2	177147
		By Hub	521441

SIMULATION



If you were at
 Black Grey Brown Green Orange Green Grey
 or 262762

and wanted to get to
 Black Red Grey Orange White Brown Violet
 or 827103

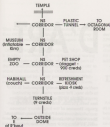
you would move to a nearby direction for the difference of 831441 (827103-262762 = 51441)

and then go to the walkway and go 8 to enter the destination.

If you were at (Merrith) 827642 and wanted to get to the exit (see BATTERED EXIT) location 0, you should start at the Hub and:

go E once (826342-827642 = -1300) and move S, 5 to the Inner ring #1 and

PLEASURE DOME

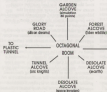


go E once (20601-20604 = 34852) then move S to the ET ring #2 and
 go E once (24852-20604 = 15769) then move S to the ET ring #3 and
 go E twice (15769-20601-20611 = 2047) then move S, 5 to the Central ring #2 and
 go E twice (2047-209-209 = 489) then go S and
 go E twice (549-249-249 = 103) then S to ET ring #2 and
 go E once (103-81 = 22) then go S, 5 to the Outer ring #2 and
 go E twice (22-8-8 = 8) then go S and
 go E twice (8-3-3 = 3) then go to the hallway and go S to exit the ETS.

Remember to gain access to your or the secretary's backdoor; you will need the relevant branch/badge. This system is very similar to Mr. Williams' but I suspect that everyone has their own slightly different system of using the ETS.

My only problem is that I don't know what I'm supposed to do, apart from getting money and points.

OVER TO YOU!



Now it's your turn to help your fellow gamblers out. If you can shed light on any game like this send us in your solutions (with maps or diagrams if you can) and IT'S GUARANTEED that it will be in a future TIPSTER column!

As always send your stuff to:
THE TIPSTER
NEW ATARI USER
P.O. BOX 94
STAFFORD
ST16 1DR

OBJET D'ART

**Joel Goodwin
concludes his
series for the
more advanced
programmer**

3. Coup de Grace

Object-Oriented Programming (OOP) became the New Big Thing in the mid 80s. Everybody had to have a piece of the action - suddenly, it was the obvious answer to every problem, but as with any new phenomenon driven by hype, a bubble in understanding which in turn meant that it became misused. Stories of projects becoming unworkably non-complex and programs running at a snail's pace started to circulate. The trouble with OOP was that it was not something that could be learnt quickly. People tried to run before they could walk and stumbled; some blamed the new programming "fad" as to more than some computer scientist's idea of a fad program. My own experience writes this as the

mathematical community responded to OOP in this way. For many years, the principal programming language for mathematicians has been Fortran, which is a procedural language. Because of the long history between mathematicians and Fortran, many software libraries of useful mathematical subroutines exist, so when OOP came on the scene there was a distinct lack of interest. Fortran was tried and tested, comfortable and the subject of recent revisions. What more could you want? The language that was being pushed by industry, C++, lacked extensive mathematical libraries and for had positive experiences with it. The former winner of the infamous "Incomparable" overhauled imposed on C++ programs made the choice clear - it was far simpler to leave well alone.

Having said all this, there is one growing recognition in mathematics towards an OOP approach. Mathematicians desire software libraries and if there's one thing that OOP is good at, then it is the reuse of software. There is still some way to go, though. I attended a day of talks at Oxford University, some time in 1986, on the subject of mathematical programming. One of the issues being discussed was which was better for mathematics - a procedural approach or an object-oriented approach? During the journey home I chatted to a lecturer about the Fortran versus C++ debate. He said it was about which was best - truth or beauty? And truth, he added, won.

Variable Name	Type	Initial Value	Default Value
SCREEN_SIZE	INTEGER	800	800
SCREEN_DEPTH	INTEGER	256	256
SCREEN_WIDTH	INTEGER	800	800
SCREEN_HEIGHT	INTEGER	600	600
SCREEN_REFRESH	INTEGER	1	1
SCREEN_BUFFER	INTEGER	1	1
SCREEN_MODE	INTEGER	1	1
SCREEN_COLOR	INTEGER	1	1
SCREEN_FONT	INTEGER	1	1
SCREEN_CURSOR	INTEGER	1	1
SCREEN_KEY	INTEGER	1	1
SCREEN_MOUSE	INTEGER	1	1
SCREEN_SOUND	INTEGER	1	1
SCREEN_VIDEO	INTEGER	1	1
SCREEN_AUDIO	INTEGER	1	1
SCREEN_NETWORK	INTEGER	1	1
SCREEN_KEYBOARD	INTEGER	1	1
SCREEN_MOUSE_BUTTON	INTEGER	1	1
SCREEN_MOUSE_BUTTON_2	INTEGER	1	1
SCREEN_MOUSE_BUTTON_3	INTEGER	1	1
SCREEN_MOUSE_BUTTON_4	INTEGER	1	1
SCREEN_MOUSE_BUTTON_5	INTEGER	1	1
SCREEN_MOUSE_BUTTON_6	INTEGER	1	1
SCREEN_MOUSE_BUTTON_7	INTEGER	1	1
SCREEN_MOUSE_BUTTON_8	INTEGER	1	1
SCREEN_MOUSE_BUTTON_9	INTEGER	1	1
SCREEN_MOUSE_BUTTON_10	INTEGER	1	1
SCREEN_MOUSE_BUTTON_11	INTEGER	1	1
SCREEN_MOUSE_BUTTON_12	INTEGER	1	1
SCREEN_MOUSE_BUTTON_13	INTEGER	1	1
SCREEN_MOUSE_BUTTON_14	INTEGER	1	1
SCREEN_MOUSE_BUTTON_15	INTEGER	1	1
SCREEN_MOUSE_BUTTON_16	INTEGER	1	1
SCREEN_MOUSE_BUTTON_17	INTEGER	1	1
SCREEN_MOUSE_BUTTON_18	INTEGER	1	1
SCREEN_MOUSE_BUTTON_19	INTEGER	1	1
SCREEN_MOUSE_BUTTON_20	INTEGER	1	1
SCREEN_MOUSE_BUTTON_21	INTEGER	1	1
SCREEN_MOUSE_BUTTON_22	INTEGER	1	1
SCREEN_MOUSE_BUTTON_23	INTEGER	1	1
SCREEN_MOUSE_BUTTON_24	INTEGER	1	1
SCREEN_MOUSE_BUTTON_25	INTEGER	1	1
SCREEN_MOUSE_BUTTON_26	INTEGER	1	1
SCREEN_MOUSE_BUTTON_27	INTEGER	1	1
SCREEN_MOUSE_BUTTON_28	INTEGER	1	1
SCREEN_MOUSE_BUTTON_29	INTEGER	1	1
SCREEN_MOUSE_BUTTON_30	INTEGER	1	1
SCREEN_MOUSE_BUTTON_31	INTEGER	1	1
SCREEN_MOUSE_BUTTON_32	INTEGER	1	1
SCREEN_MOUSE_BUTTON_33	INTEGER	1	1
SCREEN_MOUSE_BUTTON_34	INTEGER	1	1
SCREEN_MOUSE_BUTTON_35	INTEGER	1	1
SCREEN_MOUSE_BUTTON_36	INTEGER	1	1
SCREEN_MOUSE_BUTTON_37	INTEGER	1	1
SCREEN_MOUSE_BUTTON_38	INTEGER	1	1
SCREEN_MOUSE_BUTTON_39	INTEGER	1	1
SCREEN_MOUSE_BUTTON_40	INTEGER	1	1
SCREEN_MOUSE_BUTTON_41	INTEGER	1	1
SCREEN_MOUSE_BUTTON_42	INTEGER	1	1
SCREEN_MOUSE_BUTTON_43	INTEGER	1	1
SCREEN_MOUSE_BUTTON_44	INTEGER	1	1
SCREEN_MOUSE_BUTTON_45	INTEGER	1	1
SCREEN_MOUSE_BUTTON_46	INTEGER	1	1
SCREEN_MOUSE_BUTTON_47	INTEGER	1	1
SCREEN_MOUSE_BUTTON_48	INTEGER	1	1
SCREEN_MOUSE_BUTTON_49	INTEGER	1	1
SCREEN_MOUSE_BUTTON_50	INTEGER	1	1
SCREEN_MOUSE_BUTTON_51	INTEGER	1	1
SCREEN_MOUSE_BUTTON_52	INTEGER	1	1
SCREEN_MOUSE_BUTTON_53	INTEGER	1	1
SCREEN_MOUSE_BUTTON_54	INTEGER	1	1
SCREEN_MOUSE_BUTTON_55	INTEGER	1	1
SCREEN_MOUSE_BUTTON_56	INTEGER	1	1
SCREEN_MOUSE_BUTTON_57	INTEGER	1	1
SCREEN_MOUSE_BUTTON_58	INTEGER	1	1
SCREEN_MOUSE_BUTTON_59	INTEGER	1	1
SCREEN_MOUSE_BUTTON_60	INTEGER	1	1
SCREEN_MOUSE_BUTTON_61	INTEGER	1	1
SCREEN_MOUSE_BUTTON_62	INTEGER	1	1
SCREEN_MOUSE_BUTTON_63	INTEGER	1	1
SCREEN_MOUSE_BUTTON_64	INTEGER	1	1
SCREEN_MOUSE_BUTTON_65	INTEGER	1	1
SCREEN_MOUSE_BUTTON_66	INTEGER	1	1
SCREEN_MOUSE_BUTTON_67	INTEGER	1	1
SCREEN_MOUSE_BUTTON_68	INTEGER	1	1
SCREEN_MOUSE_BUTTON_69	INTEGER	1	1
SCREEN_MOUSE_BUTTON_70	INTEGER	1	1
SCREEN_MOUSE_BUTTON_71	INTEGER	1	1
SCREEN_MOUSE_BUTTON_72	INTEGER	1	1
SCREEN_MOUSE_BUTTON_73	INTEGER	1	1
SCREEN_MOUSE_BUTTON_74	INTEGER	1	1
SCREEN_MOUSE_BUTTON_75	INTEGER	1	1
SCREEN_MOUSE_BUTTON_76	INTEGER	1	1
SCREEN_MOUSE_BUTTON_77	INTEGER	1	1
SCREEN_MOUSE_BUTTON_78	INTEGER	1	1
SCREEN_MOUSE_BUTTON_79	INTEGER	1	1
SCREEN_MOUSE_BUTTON_80	INTEGER	1	1
SCREEN_MOUSE_BUTTON_81	INTEGER	1	1
SCREEN_MOUSE_BUTTON_82	INTEGER	1	1
SCREEN_MOUSE_BUTTON_83	INTEGER	1	1
SCREEN_MOUSE_BUTTON_84	INTEGER	1	1
SCREEN_MOUSE_BUTTON_85	INTEGER	1	1
SCREEN_MOUSE_BUTTON_86	INTEGER	1	1
SCREEN_MOUSE_BUTTON_87	INTEGER	1	1
SCREEN_MOUSE_BUTTON_88	INTEGER	1	1
SCREEN_MOUSE_BUTTON_89	INTEGER	1	1
SCREEN_MOUSE_BUTTON_90	INTEGER	1	1
SCREEN_MOUSE_BUTTON_91	INTEGER	1	1
SCREEN_MOUSE_BUTTON_92	INTEGER	1	1
SCREEN_MOUSE_BUTTON_93	INTEGER	1	1
SCREEN_MOUSE_BUTTON_94	INTEGER	1	1
SCREEN_MOUSE_BUTTON_95	INTEGER	1	1
SCREEN_MOUSE_BUTTON_96	INTEGER	1	1
SCREEN_MOUSE_BUTTON_97	INTEGER	1	1
SCREEN_MOUSE_BUTTON_98	INTEGER	1	1
SCREEN_MOUSE_BUTTON_99	INTEGER	1	1
SCREEN_MOUSE_BUTTON_100	INTEGER	1	1

This was rather depressing because it meant that OOP was still being perceived as a way to make the program code look elegant and little else.

Object-orientation is not about 'objet d'art', it is about confronting complexity and sorting them. Elegance of code is, of course, a part of this, a well-structured program is far easier to reuse and dissect. Today OOP is considered to be essential by many programmers. Without hesitation I must myself amongst them.

Recall that the FLAK class file made two demands on the main program: the label \$ZFLAK must be given the address of a zero page vector which can be used by the FLAK class subroutines, and the zero page locations \$R0 and \$R1 must hold the screen memory address.

DEMONSTRATION TIME

As pointed last time, we are going to look at a demonstration of the FLAK class. The beauty of object-orientation is that code developed for a class is valid for as many distinct objects of that class as we have memory for. The whole point of classes is to isolate structures from a particular implementation, to enable reuse of that structure. So to demonstrate the distinct reduction of complexity that OOP achieves, the demo program will manage 100 different FLAK objects at once. Listing 1 is the MAC/68 listing of the demonstration. Listing 2 (in this issue's disk) is a BASIC listing which will create an executable disk file of the demo program (use DOS option 'L' to load FLAKD800.CH4).

The MAC/68 code is not difficult to follow. First, the program creates a blank graphics mode 0 canvas by using %c, the screen handle. The canvas is erased and the colours are set.

Then the program deals with creating each object. Each object is given a random direction and speed and is represented by an inverse character. Each object is also given a random position to start from.

Finally, the program deals with processing the movement of each object, which just involves \$R0 \$R1, \$R0 MOVE and \$R1 MOVE. There is some baggage associated with

TAKING FLAK

Last issue, we looked at a class support mechanism for machine language programmers lucky enough to own a copy of MAC/68. We then constructed the FLAK class, an object of which represents a moving graphics mode 0 character. An object holds the following pieces of public data:

- \$POS - Horizontal position on screen
- \$YPOS - Vertical position on screen
- \$DIR - Direction of movement
- \$CHAR - Character to use when plotting
- \$SPEED - Speed of movement

In the above, \$DIR can take values from 0 to 7, representing north, north-east, east, etc. The value of 255 means the object is stationary. The public subroutines are:

- \$INIT - Initialise object data
- \$PLOT - Plot object on screen
- \$ERASE - Erase object from screen
- \$MOVE - Process object movement

An object's position will only advance after several MOVE calls: this is done so that different objects move at different speeds. The exception to this is if the \$SPEED is set to zero where the object's position will be altered on every MOVE call.

```

OBJ NO LIST                                APO 0000  ;This makes
; Declaration of FLAK class                 ORA 0000  ;ID=ORAN+100
; by Joel Goodwin 11-1-87                  OPUT CHAR
;                                           LDA 0000
;                                           OPUT SPEED ;SPEED random
;                                           LDA 0000
;                                           APO 0000
;                                           OPUT DR   ;DR random 0-7
;                                           LDA 0000
;                                           AND 000F
;                                           ORA 0000
;                                           BCS 0010
;                                           OPUT XPOS ;XPOS rand. 0-255
;                                           LDA 0000
;                                           AND 001F
;                                           ORA 0000
;                                           BCS 0010
;                                           OPUT YPOS ;YPOS rand. 0-255
;                                           JBR 0000
;                                           JBR 0000 ;rand FLAK xid.
;                                           BNE 0010
;                                           ; Wait for VBL before moving
;                                           LDA 0000 ;Wait for VBL event
;                                           BNE 0010
;                                           STA 0010 ;CPU wait
;                                           LDA 0000 ;Start timer for
;                                           STA 0010 ;start time
;                                           ; Process block objects
;                                           JBR 0000 ;FIRSTFLAK
;                                           JBR 0000 ;NEXTFLAK
;                                           BNE 0010 ;Process block objects
;                                           ; Miscellaneous data
;                                           PROCESS
;                                           JBR 0000 ;MOVE
;                                           JBR 0000 ;PLAT
;                                           JBR 0000 ;NEXTFLAK
;                                           BNE 0010 ;PROCESS
;                                           BCC 0010 ;finished block
; Main program
;
; * Initialize display
START  LDC 0000
      LDA #10
      STA SCOMM,X
      JBR 0010 ;CLOSE #
      LDC 0000
      LDA #1
      STA SCOMM,X
      LDA #SHAND
      STA SHAND,X
      LDA #SHAND
      STA SHAND+1,X
      LDA #0
      STA SCAL,X
      JBR 0010 ;GRAPHICS 0
      LDA #1
      STA TIB
      LDA #0
      STA T1,X
      LDC #0
      LDA #0
      STA DR,T
      ; * Initialize stack
      LDA #TEMPO
      STA #STLOC
      JBR #FIRSTFLAK
      JBR 0010
      LDA #0

```

moving through a block of FLAK objects (see the FIRSTFLAK and NEXTFLAK subroutines), but this is superfluous to the core of the draw which is the FLAK class itself.

The program maintains the timing of the movement loop carefully. It is possible that one cycle of movement will be quicker than the next, because many objects may not have physically moved at all. So after the FLAK object movement has finished, the program will wait to make sure each cycle takes exactly the same time. It uses the zero page timer STLOC (lines 18, 19, 20) to make sure the duration of each movement cycle is the same. Another reason for waiting after each cycle is that it is possible that certain cycles may be so quick that several cycles will have been processed before the display has had time to show one of them. The 6502 processor can get through plenty of machine language before the end of a single TV frame; by forcing the length of a cycle to be at least one frame, each cycle is always visible.

There are two parameters which can be altered to affect the program behaviour. The first parameter, TEMPO, governs the duration of the movement cycles. It can vary from 0 to 255. The program, as given in listing 1, does not use the fastest TEMPO value because the work involved in some cycles exceeds that allowed by TEMPO=0; the result of using a excessive TEMPO value is frequent pausing in the motion. The second parameter, TOTAL, is the number of FLAK objects to be managed. The program is designed to handle up to 255, but it could be easily modified to cope with more than this. Of course, the higher TOTAL becomes, the lower TEMPO must be.

Consider how you might have handled writing a program designed to draw 120 characters around the screen, each in different directions with different speeds. It is common, in such a situation, to define arrays for each

piece of data. The difference here was that we arranged the data into groups which belonged to the same FLAK object rather than, for example, separate speed and position arrays. Although the array approach has optimisation aspects to match for it, the class approach is quicker, simpler and easier to bring front on to this last end. Please bear in mind, however, that the class could have been made faster, without breaking an OOP approach, by some restructuring of the sub-routines. This has not been done as it would have made the class more complicated than was necessary for the purpose of these articles.

The OOP approach is not just better because of the development speed, though. We could take the FLAK class and apply it in a different program if we wanted, with no hassle at all. But what do we do if we wanted to extend the class?

INHERITANCE IN MAC/65

Primitive inheritance can be achieved in MAC/65 with little effort. To derive a new class DERIVED from another class ORIGINAL, there are used to use a special instruction after the NEWCLASS declaration. The first line of our class declaration must read 'CLASS ORIGINAL'.

```

NEWCLASS
CLASS ORIGINAL ;inherit from ORIGINAL
DATA1 0000
DATA2 0010
...
DERIVED INCLASS

```

Recall what inheritance means. The DE-

OF THE ORIGINAL CLASS. THE ORIGINAL CLASS'S SUBROUTINES WILL WORK ON DERIVED.

A specific example might be more instructive. Consider the **FLAK** class created last issue. We want to use the class for a game, where each **FLAK** object is a target which is worth some points when hit. We decide that it would be better to add the public data member **SCORE** to the **FLAK** class to facilitate this. Conventional programming would have us go back to the **FLAK** class file and "tack" in the **SCORE** member. GOF suggests we inherit a new class from the old using inheritance, as follows.

NEWCLASS
CLASS FLAK
INHERITS FROM FLAK
SCORE BYTE
GAMEFLAK ENDCCLASS

Our game would then use the **GAMEFLAK** class throughout. We would need to INCLUDE the appropriate files, of course - both the **FLAK** and **GAMEFLAK** class files are required and must be INCLUDE'd in that order. This example may appear trivial but this is only scratching the surface. Inheritance can prevent many more additions than just a **SCORE**. The derived class can also have its own subroutines, public and private.

Now would be the time to point out limitations in this approach. MAC/65 cannot support "multiple" inheritance - the ability to derive a class from two different classes simultaneously. The following would NOT work:

NEWCLASS
CLASS FLAK
INHERITS FROM FLAK
SCORE BYTE
GAMEFLAK ENDCCLASS

NEWCLASS
CLASS MAN
INHERITS FROM MAN
Does NOT inherit from WOMAN
CHILD ENDCCLASS

What you could do to bypass this problem is inherit from one class and then not an object

from the other class as a data member, i.e.

NEWCLASS
CLASS MAN
INHERITS FROM MAN
Does NOT inherit from WOMAN
CHILD ENDCCLASS

The **CHILD** class only truly inherits from one class. The members of **MOTHER** need to be accessed indirectly. Supposing **JOHN** is a **CHILD** object, then references to the encapsulated **MOTHER** object must be made to a **JOHN.MOTHER** address member-style. Similarly, **WOMAN** subroutines cannot be used on **JOHN**, they must be used specifically on **JOHN'S MOTHER** object. This appears to disturb the symmetry in the **CHILD** example and it would probably be better not to inherit from the **MAN** class after all, creating it as an object called **FATHER** makes more sense. The idea of nesting objects within objects is not necessarily bad, if we go back to our interesting **POSITION** class then we could define a **POLYLINE** class containing several **POSITION** objects from which a line can be described. To inherit **POLYLINE** from **POSITION** makes no sense.

Another drawback in MAC/65 inheritance concerns the private members of the original class. When deriving a new class, the new class unfortunately cannot access any of the private members from the original class. The LOCAL stack around the original class file contains the private data and subroutines invisible to the code in the derived class file. There is no way of regaining this visibility.

You may be interested in learning how inheritance works in MAC/68. Consider writing an object of one class as the first member in the definition of a new class. The data members of the old class are indexed by their labels, e.g. the first member corresponds to the first byte, the second member, perhaps, to

PLANNING ADVANCED

the third byte and so on. Because this class is created right at the start of the new class then all these indices remain EXACTLY THE SAME for the new class. That is, the first member of the old class corresponds to the first byte in the new class, the second member to the third byte and so on again. Therefore, we need not treat these data members as if belonging to the old class; we can pretend that they were part of the new class all along. The inheritance falls out naturally.

POLYMORPHISM IN MAC/65?

Polymorphism is a lovely idea but can we make it using MAC/65? No. Polymorphism, for our purposes, means that classes derived from a common class can inherit a sub-routine call, but replace the actual sub-routine.

Suppose we derived two different classes from the **FLAK** class: **DEADFLAK** and **LIVEFLAK**. The **DEADFLAK** objects do not move and the **LIVEFLAK** objects move changing their direction frequently. Polymorphism would allow us to implement new **MOVE** subroutines for both **DEADFLAK** and **LIVEFLAK** which would mean the main program would never have to determine whether it was dealing with a **DEADFLAK** or **LIVEFLAK** object, all the program would need would be **JOB MOVE** and the correct sub-routine would be called automatically. No, we cannot do this. This is not so bad because polymorphism, in some GOF languages, must be pre-planned. That is, we should have declared the **FLAK** class sub-routine **MOVE** as a candidate for polymorphism from the outset.

What we can do is create a data member

which holds a sub-routine address. Instead of a set **MOVE** sub-routine, we have a **MOVE** data member:

NEWCLASS
MOVE BYTE
FLAK ENDCCLASS

The main program would not use **JOB MOVE**; it would have to pull out the two **MOVE** bytes and create the correct **JOB** instruction while moving. A menu could be created to do this, call it **MOVE**, so the main program could simply state **MOVE MOVE**. Note that the **MOVE** subroutines can be different for two objects of the same class. If this was true polymorphism, then **MOVE** could only be different for objects of different classes. But then again, you can't have everything. This row wraps up everything I wanted to say about using classes in MAC/65 projects. So where do we go from here? What do we do with this new potential?

POTENTIAL

Many program agents find themselves very well in GOF methods. When I wrote *Mathematics (G&A) Issue 78*, I had object-orientation in mind even though *OS/BOOTS.M65* hadn't been developed there; player status graphics are ideal for applying GOF ideas to. Look at the structure that the Basic programmer sees. Each player has several inputs: **IMAGES**, **HP**, **XP**, **YPOS**, **SEX** and **COL**. Many of the visible aspects of PMG are removed and unappreciated to the VM, it even destroys the identity of each instance, combining them into an independent player almost seamlessly. Private data belonging to each

player is hidden from the user; Mutation correctly presents the vertical position and length length used by the last VBL. This is what OOP is all about. We cannot see any of the machinery and we never need to. Mutation, of course, is not perfect as I stated in the accompanying article. There is plenty of scope for improvement, to strengthen the object-oriented aspects.

A project which cries out for an OOP approach is a windows graphical-interface system. Each window is perceived as independent entity; why not develop a windows program in the same way by using OOP? Think of what data and subroutines should be attached to a WINDOW object. Are there different types of windows? Would subroutines play a part? How can we involve a "pointer" to interface with a window? And so on.

One problem I've faced when doing extensive machine language work is that when graphical changes are issued they can sometimes end up out of sync. Suppose I wanted to change the entire screen display; all of the player-visible graphics must change, a different display list will be selected and new display list interrupts will be needed. If I simply programmed up these changes one after another then it is likely that there will be a brief flash of chaos as some of these changes become visible before all of the changes have been finished. The easy way out is to turn SYNC off with the POKE \$9A,0 trick, do all the changes, then turn it back on. This works but cannot give the impression of shifting from one situation to another continuously. The brief black screen is a pain in the eye. I've always wanted to be able to change the entire screen display without such an interruption. All of the changes can be delegated to the VBL to accomplish this, but it gets very complicated, very quickly. OOP helps because by constructing a SCREEN class consisting of all

the important screen data, the information is centralized and the situation is controlled.

We have examined just three situations where OOP can help. There are many, many more. Some of these are not obvious but all are effective in reducing complexity and improving maintainability.

CODA

It has been a long haul, but it is finally time to conclude our discussion of OOP. The aim of Object First was to consider some new programming ideas to the reader and to discover inside their benefits. This is not easy to do in just three articles. The principles of object-orientation have been explored far more thoroughly elsewhere; what you have read here is merely a gross simplification of OOP. There are plenty of books on the subject, but rarely I have found many of them accessible to the casual programmer. Most are aimed at the three tiers in modern programming terminology and normally experienced in the language C.

Nevertheless, I hope I have stimulated your interest in something different. We have to remain aware of what is going on around us as the world moves on and there is always more to learn. Just as object-orientation has been established as the de-facto and real-ill, its flaws are already being examined. OOP is not the "ultimate" approach. It is just the latest one. New paradigms are invented every day - for example, have you ever heard of polytype programming? Who knows what the next revolution is or what it will bring? We must all keep our eyes open and be prepared to accept new ideas - otherwise we will quickly become nothing more than a collection of interesting antiquities.

XL/E PROGRAMMING

HAPPY TYPER

by Steve Hillen

*Dennis Fogerty
revives one of his
favourite programs
from Monitor
magazine*

Long ago when I did a lot of programming and typing of listings on my XL I found this program a gem. Today, hopefully, others will find it helpful too. All credit must go to Steve Hillen, whose are you now?, and I hope he won't object to my re-publishing his article here verbatim.

INTRODUCTION

The Happy Typer is a utility for use with Atari Basic. It will give you automatic intelligent line-numbering and 18 extra keys which you can reconfigure to print out keywords, thus speeding up your typing. Unlike many auto-line numbering facilities, this one allows the full use of the Atari screen editor, so you can adjust lines while still in the auto mode. The redefined keys are accessed by pressing the SHIFT and CONTROL keys simultaneously with a number key. These keys are not used by Basic or the operating system, so you can still type in all those needed characters.

MAKING A BOOT DISK

For a disk system, type in the listing and save it. Type RUN and the program will check your typing and ensure that the data is correct. Re-type those lines that produce an error. Once it is ready the program will ask you to insert a disk with DOS on it. The program will then save out an Autostar .app file onto the disk. Don't change the filename - Happy Typer will only work as an Autostar .app file. The next time you boot this disk with Basic, Happy Typer will be ready for use.

USING HAPPY TYPER

THE AUTO LINE NUMBERING

Every time you tap the TAB key after a RETURN, a new line number will be printed. If you type on the TAB key and the last key pressed was not a RETURN then the TAB will be performed as normal. This is better illustrated by example.

Directly after power-up, press the TAB key - the first line number will be printed. Type "Hello" then RETURN then TAB. The next line number will appear. Play around and get used to using the TAB key after a RETURN. If you type in a few lines of Basic, then hit them, then press RETURN and TAB, the next line number after the last line of your program is printed. Also, if you use the cursor

CD COLLECTION DATABASE

This CD collection database program was written by me in 1997 in an attempt to catalogue my ever-growing collection of CDs.

The first question you're probably about to ask is why anyone would actually want to catalogue their CDs? Well, there are a number of reasons for this. Firstly, it gives you a hard record of exactly what you have (and, let's face it, when you've got more than about 50 CDs, you're unlikely to be able to remember them all). This would be useful in the event of an insurance claim, should your collection be stolen or damaged. In my case, I have around 600 CDs, many of which are hard to find or limited edition versions, and are therefore of great monetary and sentimental value to me. Incidentally, it's well worth keeping receipts and taking photos of your collection regularly in the event of an insurance claim arising.

I also find the program useful because it allows me to search quickly for all of the CDs I have by any one artist, (useful because I sometimes sell CDs). If you wanted to put a number after each CD name, in brackets, you could use the program to tell you where in your collection to find the particular CD you're looking for. This feature is quicker than looking through all CDs by hand, especially as the writing on the spine of a CD case is usually small, in different fonts, and sometimes with the artist and sometimes with the title appearing first.

by Kevin Cooke

FIRST GET A WORD PROCESSOR

This relatively short program, written in Turbo BASIC (13 years old and still one of the best versions of BASIC I've ever encountered), is simply a highly-specialised text editor. The four databases are stored as text files (created with a word processor of your choice) and then read by the program. You may criticize my choice of making you use a separate word processor to create your database, but let me explain my reasoning. Firstly, any text text editor program I wrote would have difficulty competing with the flexibility of a dedicated word processor. Secondly, despite the speed of computers, when you've got a large file of information, nothing can sort things into alphabetical order faster than YOU! It seemed silly to write a program to sort each new entry into its correct alphabetical database position when it's so easy to do it yourself! Thirdly, I wanted to keep the program as short as possible so that it would load and run quickly - if it didn't do this, it would defeat the whole purpose of the program, which is to save you time.

OPTIONS

When the program loads and runs, you are presented with a screen offering you 7 options. The first two allow you to search for all titles by the artist of your choice to either the album or single databases. The next two options allow you to view all singles or albums in the database in order, in case you can't find the artist that you are looking for in your search using the first two options.

The next two options allow you to look at the database of the music compilations and movie soundtracks stored on the disk. Finally, the last option quits to DOS.

SAMPLES

On this issue's disk you will find the databases ready to run along with four sample databases (text files).

- ALBUMS** - The album database
- SINGLES** - The single database
- COMPILAT** - The Compilation database
- SOUNDTRA** - The Movie soundtrack database

The format of each database file, as you will see by examining my sample files, must be as follows:

For the albums or singles database:

- SURNAME, FIRST NAME
- list Albums or single title
- list Albums or single title
- list Albums or single title

- *
- NAME OF BAND/GROUP
- list Albums or single title

- WCT WCT WCT
- Picture this
- End of part 1

- *
- END OF FILE

For the movie soundtrack or compilation database files:

- MOVIE/TITLE COPY #
- *
- MOVIE/TITLE, YEAR
- *
- MOVIE/TITLE
- *
- END OF FILE

The easiest way to keep in this format is to copy the sample database files, along with the main program, onto a separate disk and edit the sample files to contain the titles of your own CDs.

As can be seen, for the movie soundtrack and compilation databases, you do not have to enter the artist name. This is because the tracks on these CDs are usually by a number of different artists, hence the program automatically displays "Various Artists" under the "Artist" heading when the file is viewed.

TAPES TOO

Of course, I mean no prejudice towards cassette tape users by referring to CDs throughout these instructions. The program is equally as suitable for people who only own cassettes, but I am recognizing the fact that nowadays a lot of people have CD players and few shops have a wide selection of cassette albums on cassette tape.

I hope you enjoy using the program. It fills my needs, and I hope it fills yours. If you have any suggestions or comments, feel free to write to me via Page 6 to let me know.

THE LISTING

This program can be found on this issue's disk ready to run. A TTPC coded type-in listing is available on request - see inside back cover

USEFUL USR

John Foskett
explains the easy
way to connect
machine language
with Basic

There is no doubt that USR is a very important instruction in the BASIC language since this is the only instruction with which the BASIC programmer can obtain the advantages of machine code. For instance it is often necessary to store a block of data from one address to another such as when copying the character set into RAM, moving PMOS vertically, etc. Doing this in BASIC can be very time consuming but by using a machine code routine via USR, the process is virtually instant. The advantages of machine code are obvious - much reduced calculating time, much PMO movement, automation, etc. and all this is available to the BASIC programmer via USR. Basically, what is awkward and difficult if not impossible in BASIC is a breeze in machine code.

WHAT IS USR?

The USR instruction is used to GOOSE to a machine code subroutine after which control is returned to BASIC in exactly the same way as BASIC goes to a BASIC subroutine. It is similar only to passing values into a BASIC subroutine, that is by equating variables prior to calling the subroutine, values or parameters may if necessary be passed into the machine code routine via the USR call. When used without additional parameters, the USR function contains only the address of the machine code routine thus (assuming the routine is at the start of page 0)...

`USR(0)`

Alternatively, the machine code routine may be contained within a string as a relocatable code (say MCH) in which case USR is used as follows...

`USR(MCH)`

In some cases when the machine code routine is relatively small and particularly if only used once, the machine code routine representing MCH may be placed directly into the USR function as follows...

`USR(ADR("machine code string"))`

When USR is used with additional parameters, they follow the address of the machine code routine and separated by commas. Assuming the use of TWO additional

parameters (A and B) the above three USR functions become...

`USR(ADR(A,B))`
`USR(ADR(MCH),A,B)`
`USR(ADR("machine code string"),A,B)`

THE STACK

When running any program, the computer constantly uses the stack (page 1 of RAM) as a temporary storage for storing addresses and values when actioning subroutines, loops, etc. The stack is a special part of the computer RAM which requires no addressing and no reference of any kind, information is simply stored there, where it is actually put is immaterial. The best way to visualize how the stack works is to imagine a pile of plates where the plates must be put on and taken off the pile in order, thus the last plate to be removed, must be the first plate to be removed. From this it can be seen that the stack works on the "Last In First Out" LIFO principle and so long as this rule is strictly adhered to, there will be no stack errors. This then is the reason why no addressing or referencing of any kind is required in order to use the stack.

WHY MENTION THE STACK?

Normally a programmer doesn't need to consider the stack since its working is fully automatic via the computers operating system. The only time the BASIC programmer must consider the stack is when writing the machine code routine which will be actioned via the USR command. This is because USR stores at least one value on the stack during its operation which MUST be removed prior to

returning to BASIC in order to let the computer "see" the return address which it stored there prior to actioning the routine in the first place. If this is not done then it can be seen why the computer may produce stack errors (error 10) or even crash or lock up since it will obviously read the wrong return address prior to returning to BASIC.

THE PARAMETERS

In order to pass any parameters to the machine code routine that the USR command may contain, USR places them on the stack following the return address in a two-byte format. USR stores the low byte first followed by the high byte and in the order of the last parameter first which allows them to be retrieved from the stack to be an input response which is the first parameter to the USR call first in the order of high byte first.

THE ODD BYTE

After all the parameters (if any) have been stored on the stack, the USR command stores a further value on the stack, a value of its own, a record of the number of parameters included in the call. This odd byte from now on referred to as the "odd byte" may at first appear absolutely useless, but it can have a very important role to play.

THE MACHINE CODE ROUTINE

As previously stated, all parameters and the odd byte must be removed from the stack by the machine code routine before exiting back

then a crash will result. The routine is called in the following two ways: X=USR(1504,2) to set the screen colour and X=USR(1506) to reset it back to blue.

```
LDX #148
PLA
BEQ RESET
PLA
PLA
TAX
RESET
STX #10
RTS
```

The first instruction in this routine is to load the X register with the number 148 using LDA Load X register which is the default value for the normal start blue screen. The next instruction is the PLA which removes the odd byte from the stack and since this routine allows the use of only one or no parameters to be used in the USR command, the next instruction, the BEQ branch if equal (to zero) is used to determine in which mode the routine is being used. If the value was found to be zero, that is the USR call was made without a parameter, then the routine branches to the label RESET. The default colour value of 148 originally loaded into the X register at the start of the routine is then stored in location 710 using STA before X-register after which the routine is exited using RTS. If however the odd byte was not zero (in other words it was 1), that is the USR command was made with a parameter, then the parameter is removed from the stack using the two PLAs and its low byte loaded into the accumulator. The colour value is then transferred into the X register using TAX converting the default value of 148 originally put there. The value is then stored in location 710 using STX after which the routine is cued back to BASIC via RTS as before. Also note that PLA does not have to be the first instruction in the source code.

ROUTINE 5: RETURNING VALUES TO BASIC

Source code for screen USR01.BNC
USR01.BNC
BASIC demo for screen USR01.BAS
USR01.BAS

The variable used to store a machine code routine when using USR (X) in these examples can be used to return values to BASIC after the routine has been executed. The value loaded into the BASIC variable is determined by the two zero page locations 212 and 213 in the normal low/high 2-byte format. The following routine works in the opposite mode to the previous routines in that this routine randomly selects the colour value itself, changes the screen colour accordingly and returns the value to BASIC so that you know what the selected value was. The routine is executed using X=USR(200) where the colour value is loaded into the variable X. The routine may also be altered using "PRINT USR(1000)" or "Y=USR(1000)" which will print the selected colour value directly on screen.

```
Listing 1 (USR01.BNC)
PLA
LDA #50776
STA #10
STA #10
LDA #0
STA #10
RTS
```

```
Listing 2 (USR02.BNC)
PLA
STA #10
LDA #50776
STA #10
STA #10
RTS
```

In the first listing, the first instruction is the

normal PLA which removes the odd byte from the stack. Next the processor's accumulator is loaded with a random number from location 50776 (the random number generator) which is in the range of 0 to 255 inclusive. The random number selected is stored in the colour register 710 as before and also into location 212; the low byte of the value for returning to BASIC. The accumulator is then moved by loading it with the number zero which is then stored in location 213; the high byte of the value for returning to BASIC because the high byte will always be zero. The routine is then exited using RTS as before.

The second listing shows a little trick that can be done by re-arranging the listing to reduce the length of the routine to save memory (2 bytes in this case). Because no parameters are used when calling this routine, the odd byte will automatically be zero which will be loaded into the accumulator by the PLA instruction. Since the accumulator contains a zero at this point, it makes good programming sense to take advantage of it and use it to zero location 213 so that "LDA #0" may be omitted from the listing and so reduce the length of the final routine.

Note that there is a bug in the Turbo BASIC compiler which prevents USR from returning the correct values to BASIC. If a program using this feature is to be compiled, then the two locations 212 and 213 could be PEEKed from BASIC in the normal way and then the two values combined using the usual low/high two byte calculation as follows...

```
X=PEEK(212)+256*PEEK(213)
```

A MEMORY MOWER ROUTINE

Source code for screen USR03.BNC
BASIC demo for screen USR03.BAS

To conclude this article, a very useful memory eating routine which works to a similar

way to Turbo BASIC's MOVE command. The routine will move up to 255 bytes of data from one address in memory to another. The routine may be used in conjunction with PRINT to create smooth vertical movement as demonstrated in the BASIC demo program. The routine is executed using...

```
X=USR(1508,ADDR1,ADDR2,NUM)
```

Where NUM number of bytes to moved from address ADDR1 to address ADDR2.

```
PLA
PLA
STA #04
PLA
STA #08
PLA
STA #04
PLA
STA #08
PLA
STA #08
LDY #0
LOOP
LDA (NUM),Y
STA (NUM),Y
INY
CPY #07
BNE LOOP
RTD
```

Note the necessary use of the zero page locations 003 to 009 for addressing the two addresses using indirect indexed addressing. Also note that location 007 is used to store the number of bytes to be moved, but this location need not reside in page zero; it may in fact reside anywhere in memory.

The CLASSIC PD ZONE



FAREWELL TO FUTURA

THE FINAL ISSUES

This issue we present the concluding half of this examination of the FUTURA disk magazine which is now available from the Page 6 Library.

FUTURA SIXTEEN

This articles include - **ATARI 8-BIT NEWS** - Other Atari 8-bit supporters, **SOFTWARE SCENE** - Kevin Cooke reviews Turbo Riddles, DTP ATARI8 - Eric Donawick continues his column on Atari 8-bit desktop publishing. This time he looks at Dasy-Gal E, ANSWER A MOUSE - Part 5, Multi-Mouse II (Bowl Games), HARDWARE WAREHOUSE - Atari DOS Disk Structure, THE ATARI 8-BIT DOCKORHELP - Booklets Part 6, B-C, and Appendix A, **SOFTWARE SCENE 5** - Kevin Cooke returns with a review of Demo Master Update, VCS FUTURA - cartridges by Atari (C22810) - C22819, THE BLACK AND RED - Introduction and News, Jaguar CD-ROM

by
Austin Hillman

drive and game tips, Cosmos Paddler reviewed by Michael Clowrthy, Riddles review and game tips by Michael Clowrthy, Tronsep 2000 reviewed by Dan Everstock.

The programs include FERS - An extremely addictive game which can be played with an ST mouse or joystick, ULTRA TRANSLATOR - The ultimate translator for your BLUEN II can even be used with 480/800 cartridges.

DOB WELDER - Is a superb-disk analyzer which must be loaded with a translator. Use Ultra Translator, simply press SELECT at the title screen and then load DOB Wizard. Please read the documentation file before using this powerful program. Using DOB Wizard you can gain a detailed analysis of single assembly disks, recover files mistakenly deleted, display complete sector maps or technical notes, etc. MAGNIFY - An impressive screen dump utility with extra features. Try it with the two pictures, BOX and CORNER, which come with Ping.

FUTURA SEVENTEEN

This articles include - **ATARI 8-BIT NEWS** - J.P. Software, Micro-Discuss, ANSWER POINT - Kevin Cooke reports from the show of

the year, **THE ATARI 8-BIT DOCKORHELP** - Atari Adventures, a book review by Kevin Cooke, **DTP ATARI8** - Don-Magel **TURBO BASIC FLYER** - Ron Fitzer's programming column, **ADDING A MOUSE** - Part 6, **DOB Demo**, **PROGRAMMER PROFILE** - Lazzellian David Lewis, **THE BLACK AND RED** - The Jaguar CD-ROM package reviewed by Michael Clowrthy, **Disk Dudes** review and level codes, **VCS FUTURA** - Cartridges by Atari (C22810) - C22819, **ATARI 8-BIT TRIVIA** - Kevin Cooke issues a trivia challenge!

DISMAYOP - from the French disk Console News, Boot with BASIC and the desktop will auto-load. Use a joystick for the arrow keys to move the pointer to D.1 and press the joystick button for Betward. A list of all the files on the disk will appear. Select a file to load by moving the pointer over it and pressing the joystick button. This desktop will load both BASIC and machine code files. Not quite up to the standard of Windows 95, but not bad for the humble 8-bit.

CHOMP THE ROBOT - A colorful, challenging arcade game from Console News One. You must guide Chomp the Robot down the screen, rescue the stranded humans and return to the mother ship, AMAGAMM - A utility which compares programs, of up to six letters. From Console News Six, **OTHELLO BLITE** - A review of the popular board game also from Console News Six. Use a joystick to move the pointer and press the button to flip a piece. BSA 73 - Version 3 of a BASIC (in all senses of the word) word simulation game written by Alan Hirsch, set in the days when we still had a monoglyph industry.

DISCOVERY 1.2 - Analyzes the structure of your BASIC programs. Docs supplied. **PROG-**

RAM HELPER - tides down your BASIC program by converting constants into variables. Docs supplied. **FMT** - A very useful utility which allows you to format with or without DOS files. Works with the desktop. **PLAY L28** - Good demo. **PHAM APACORAM**, **DISCMASS** and **GRANDOR** are programs described in the Turbo BASIC Flyer column.

LINEUP - Comes from Clate and describes itself as the ultimate version of Turbo in the public domain, who am I to argue with that description. **THE BETER REALITY MORGANMO** - This is a 8-part microgame which really shows the power potential of a well programmed BL/ZE.

FUTURA EIGHTEEN

This articles include - **ATARI 8-BIT NEWS** - ACPC, Unified, The Page 6 50 Library.

HARDWARE WAREHOUSE - including a light pen, **THE ATARI 8-BIT DOCKORHELP** - Kevin Cooke reviews Joy Duty Dyes of Space, **NETWORK TO THE WORLD** - Communications article by Joe Haines, **DTP ATARI8** - Dasy Don 18, **FRICMAN FOR THERAPY** - Kevin Cooke looks at modern medicine, **THE TURBO BASIC FLYER** - by Ron Fitzer, **FRIGGELLY TROOP ON THE ST** - by Lee Wager and Stuart Murray, **ADDING A MOUSE** - Part 7, **The Brantley Editor**, **A SHORT HISTORY OF COMPUTERS** - What came before your Atari 8-bit, **CALLING ALL NOSTALGIA ENTHUSIASTS** - Some light humor.

LYSTER - This is a program for printing out your program listings complete with all of the special characters. There is an example listing

PD ZONE

provided for you to play with. **GAUSSIAN 3.0** - This program, from the old Atari User (Vol. 2 No. 3), alters DOS 3.0 to put GAUSS in the shadow BASIC under the OS in XL/XE machines. After the first DOS call which runs from the disk as usual, all subsequent calls will run from the OS with no loss of economy as it is automatically saved.

DEARWIS - A simple but pleasing graphics demo showing the goal of Atari 800. **PIZZER (PROGRAM)** and **PROGRAM** are the programs to be used with the Turbo BASIC Pizer column.

THE GAMES-OF-CHESS - Is a revised version of Robert de Letter's game which was published in SAU 78. **CHESS'S BEYOND** - Is his all new sequel with loads of puzzle action!

FUTURA NINETEEN

This article includes - **ATARI 8-BIT NEWS** - ACPC, URANUS, ADDING A MIDGE - Part II, Naughts & Crosses, DTP ADRES - Digital Editor V3.0, **SOFTWARE SCENE** - Bambi Taskmaster, **SOFTWARE SCENE 2** - Jew-Jewler & Mousekatch, **HARDY HERT** - Bantick XL - A SIGHT AND STUPIDITY - Light Insects, **YCS FUTURA** - 2600 Audio Modification, **PD SOFTWARE SCENE** - Megalux, **HARDWARE WAREHOUSE** - Nintendo Controls on the Atari 8-bit, **THE ATARI 8-BIT BROTHERS** - Advanced Programming Techniques for your Atari, **THE BLACK AND RED** - Atari: The Code!
NOUGHTS & CROSES - New software from Kevin Cook. See his column **ADDEND A MIDGE**, **CARPOUS BLENDING** - Colourful artwork by Kevin Cook, **REMYANTIK** - A

keyboard enhancement utility, with many functions. Full docs are included.
POST MASTER - Allows fast printing of program listings, including special characters. There are lots of options available. **NOB WAYLDOC** - A test file for use with the sideways typecast print system. Two fonts, **KOMAM** and **MOGREN**, are included for use with Options 3 and 4. **LOVE MY ADAM** - A superb singing music and graphics demo by Philip Fene and Gary Colleton. **LABEL PIV** - Allows you to read and write identity tags to your disks in the otherwise unused sector 120. **FINANCIAL CALCULATION PROGRAM** - Office help with your investments and loans, works out every difficult calculation in a trice.

SADOLEMAN is a very enjoyable game from France. It was programmed for Atari France in 1985 as part of a retrospective contest in association with Let's Japan. It was never sold and is therefore in the public domain for us all to enjoy. There are five parts to the game. Use keys 1-5 to select a part. Press **START** to play and **RESET** to return to the main menu. Great fun!

FUTURA TWENTY

This article includes - **ATARI 8-BIT NEWS** - Page 6, **Telegames**, **DTP ADRES** - Page Editor V3.0, **SOFTWARE SCENE** - Address Database and Envelope Printer reviewed by Kevin Cook, **YCS FUTURA** - 2600 video Modification, **THE BLACK AND RED** - Jaguar news, reviews, tips and a look at Atari's Enterprise unit.

OPERATOR 051 - A powerful menu utility. **MICRODOS 051** - The classic menu utility for loading machine code files. **80-8 80F** - **PS RAMDisk 2.00** and **PS 80FPower 1.20** for use with **SPARADON 3.04**, **FORMDISH** & **FORMDOS** - Format, DOS, DUP, RAMDisk all in one operation. A very useful utility by the late Alex Pignato. **SETUPDOS** - A reloaded BASIC monitor to patch DOS 3.0 for two 80 RAMDisks on a 1540 XL.
FALCON DEMO - A top-notch demo by Harold Press the specialist at the smaller to progress in the main demo. **COLOUR DEMO** - Katherine's demo. **BORDER** - 20 demo loans on our current **IMAGINE** - A lightning fast graphics demo written in Action! **KEYBOARD CENTER V3.0** - Keyboard codes utility. **AIR PUCKY** - Fast sports game based on the old master favourite. Great fun to 2-player mode! **WEDDMASTER** - Fun game, just set the files and keep the cheat options away. Easy to play.

FUTURA TWENTYONE

The regular text articles cover in a conclusion in this final issue. **ATARI 8-BIT NEWS** - Of Hardware and Console still going strong. **DTP ADRES** - Parts 7 & 8, individual art and printing progress and series conclusions. **HARDWARE WAREHOUSE** - Re-loading printer releases. **SOFTWARE SCENE** - Drive Males and Drive T. **VTC** - How it works. **YCS FUTURA** - 7800 video modification. **THE BLACK AND RED** - Jaguar news and Power Drive Italy review.
CREATE A FONT - A character set editor can be used with the Character Sets in an In-Sight article. **DECODE** - A hexadecimal type

quote game by Frank Walters. **INSTANT CHAMP ACTION NETS** - Article and a set of programs to convert alternative character sets into settings for instant loading when used with BASIC programs.

DECLARER 2.2 - Disk labeler plus printer driver. **COLOS** - An 80 column format ready to read, with docs in 80 column format ready to read.

THE LAST WORD FROM STUART

Here's what Stuart said on his farewell - "The time we find ourselves at the final page in the final Futura. I don't see this as a goodbye but more of a see-you-later. I'm always going to be around on the 8-bit scene and will probably appear now and again with an article or review in one of the Atari publications. I'm looking forward to writing a series of articles on the Atari World Video Game Championship of 1992/93. It's about time I wrote about all that happened behind the scenes!"

I'm really looking forward to getting stuck into 8-bit titles such as *Konami*, *Star Riders II*, *Ultima II*, *Aggravator*, etc., etc. There are so many titles I have purchased over the years that have hardly ever seen the inside of my disk cabinet. It is nice time for me to enjoy them all.

All that is left to say is a final gigantic **THANKS** to everyone who has supported Futura over the years! You truly are an amazing group of people! Keep using that 8-bit and remember that together, WE are Atari! See you L8-er!"

SORTING AND SEARCHING

Daniel Yelland
presents routines for
you to use in your
own programs

SORTING

A first sight sorting and searching does not appear to be a very useful thing to do when programming, however a lot of programming problems can be solved with sorting and searching techniques.

A computer game is not the application you would immediately think of for using such techniques, but what about high score tables? In football manager games and adventure games searching for details is important also. More serious applications of sorting and searching could be in a database or accounting program. This article is about some of the techniques used when sorting and searching data.

Some search techniques require the data to be sorted which explains the relationship between the two processes. The examples given will all involve numbers but could equally well be used on alphanumeric data. Also unless stated otherwise the data is being sorted into ascending order.

SELECT SORT - The most obvious sort algorithm is the one we sometimes use when sorting data. We look through the data to find the smallest element and place it in position 1, then we repeat the process for the next smallest element and so on until the data is sorted. This technique is called the selection sort and an example program demonstrating this is given in **SELECT.DAT**. You will find the program on this issue's disk. This is how it works - given 4 numbers, say

10, 7, 5, 10, 9

we look through the list at each element in turn until we find the smallest. In this case it turns out to be 5, so

5 is put in the beginning of the list so the data now looks like this

5, 10, 7, 10, 9

We repeat the process, now ignoring the 5 as we know it is in order. The next smallest number is 7 and so that is placed in the list

just after the 5

5, 7, 10, 10, 9

The process is repeated until the list is sorted

5, 7, 9, 10, 10
5, 7, 9, 10, 10

This technique is the most obvious but it is not the fastest or most efficient, so other techniques have been tried and tested.

BUBBLE SORT - Most people have heard of the bubble sort, it is probably the most famous of computing sort routines but it is also one of the worst. In this technique neighbouring pairs of elements in the list are compared and if the element higher in the list has a lower value than the element lower in the list the pair are swapped. The whole list is compared to this way with swaps taking place until the list is completely sorted. Usually a flag is set to report no swaps were made during a pass, which signals the list is sorted. We will use our original data, so

10, 7, 5, 10, 9

First Pass
7, 10, 5, 10, 9 10 > 7 so swap
7, 7, 10, 10, 9 10 > 7 so swap
7, 7, 10, 10, 9 10 > 10 so no swap
7, 5, 10, 9, 10 10 > 9 so swap

The list after first pass is

7, 5, 10, 9, 10

The largest value (10) has "bubbled" to the top which is where the technique gets it's name from.

Second Pass
7, 5, 10, 9, 10 7 > 5 so swap
5, 7, 10, 9, 10

A COUPLE OF THE BEST

5, 7, 10, 9, 10 7 > 10 so no swap
5, 7, 9, 10, 10 10 > 9 so swap
5, 7, 9, 10, 10 10 > 10 so no swap

List after second pass -

5, 7, 9, 10, 10

In this case the list is sorted after two passes. It should be noted that 3 passes are made as the process doesn't terminate until there are no swaps. **BUBBLE.SAR** on this issue's disk is an example routine showing this technique.

INSERTION SORT - Another famous one, this technique is one of the most efficient but also one of the hardest to code. It is done the way people sort a hand of cards. For example consider the hand of cards dealt as - K 5 2 9 8

Sort

K 5 2 9 8
K 5 2 9 8
2 K 5 9 8
2 5 9 8 K
2 5 9 8 K

This is done by looking at the list from left to right and "inserting" elements which are out of order. On each pass we compare with the items on the left, until we find one larger, or reach the start of the list. The others "slide up" to make way for each moved element as in the example above. We insert the current item in the correct place and repeat this up until the last item. The program **INSERT.SAR** demonstrates this.

SHELL SORT - This routine could be used to be a variation on the bubble sort. Invented by Donald Shell, in this method elements of a fixed gap apart are compared rather than adjacent elements as in the bubble sort. The

PROGRAMMING

BUBBLE SORTING

elements of the float gap apart are sorted in the same way as the bubble sort and then the gap is halved and the process is repeated. This is done until the gap is equal to 1 where a standard bubble sort then takes place.

The advantage of this routine over a standard bubble sort is that elements get to their correct position in the list quicker than in the bubble sort as elements travel further towards their correct position in each swap.

An example of this routine is in **SHRILL.BAS**

lines 2 (7)
lines 3 (2)
lines 4 (10)

not equal to 10 so continue
not equal to 10 so continue
equal to 10 so stop
"FOUND" is returned.

If the whole list is searched and the element not found we stop the process and "Not Found" is returned. An example program of this is shown in **LENNAR.BAS**.

BINARY CHOP - This is a very efficient routine in that any element of 1000 elements can be found in just 10 passes. This routine however requires the data to be sorted first.

The routine works by using pointers which can be represented as variables in BASIC which point to the beginning of the list (e.g. array) and the end of the list. These will be referred to as the "start pointer" and "end pointer" respectively. Assuming the list is in order a middle value is taken from the list and compared against the search value. If it is equal to it the process terminates and "FOUND" is returned. If the middle value is less than the search value the starting pointer is set to the data item after the middle value in the list. If the middle value is greater than the search value the end pointer is set to the data item before the middle value in the list. After the pointers have been adjusted the process is repeated until either the element is found or the pointers coincide or cross one another (e.g. end pointer points to an element below the element start pointer points to.) **BINARYCHOP.BAS** illustrates this technique. The sort routine used to sort the data is the same one as in **SELECT.BAS**.

There are many other sort and search techniques and the ones shown here are only a few moderately well known ones. Hopefully these routines will be of use in your own programs.

Since all of the data to be sorted is in lists the best data type to use to hold the elements is an array. Each element of the array holds an element of the list. Variables are used to hold the elements being compared and loop structures are used to provide the iteration of the routines (e.g. continue until list sorted).

PROGRAMMING THE TECHNIQUES

SEARCHING

LINEAR SEARCH - Again the most simple search routine in the one we ourselves use when searching for data. We look through the list until the element is found and stop when we either find the element or we have looked through the whole list.

So looking for 10 in the original list would result in:

Original list - 95, 7, 3, 10, 9

lines 1 (10) not equal to 10 so continue

THE ACCESSORY SHOP

A COUPLE OF THE BEST

#287 - DISK DOCTOR

Most advanced users may find this software invaluable for when something goes wrong or if they need to back up their disks or create their own protected software. None of the best disk utilities around include SYSDISK which examines the Volume Table of Contents and allows you to fix problems, especially when you get a disk that shows lower free sectors than you should have. There's work only in single density though. TRACKER is a most comprehensive disk utility package with the best graph facilities even seen on this type of utility. DISK and COPY viewers, search for files and words, much more in general utility. If you need to create floppy or hard sectors then PUNCH will do it for you with ease. Another sector utility is RECFIX which lets you edit, copy and duplicate sectors and much more with some extra special facilities especially for advanced programmers. Also allows you to create "data" sectors and has a built in letter speed checker. DISK* is a best utility package that will allow you to create your own custom disks by copying files, tapes to disk, boot files and more. It will also copy Multi-boot and Disk C. More programs and has comprehensive DOS utility features. To round off CDD CREATOR: SYSTEM is a version of the old CDD PROGRAM which seems to be specifically for creating ROMs and tapes that have problems on the DL.

D8#112 - PAGE EDITOR

This DISK file, One of the best, if not the only, page layout system for the Atari 8-bit is given you EVERYTHING that You like in What You Don't need pop-up editing. Page Editor is an easy to use text and graphics editor that features an 80 column display and 40 line graphics on the same screen with the ability to place text and graphics anywhere you desire. The Page Editor software shows you exactly how the page should appear on your printer (requires Epson compatible). In addition to the Page Editor program the software package includes utilities to convert word processors and files. Print Shop format clip art and additional character sets. The program can be run in Turbo mode for extra speed. The main program, together with character sets and sample pages are included on the main disk with extensive documentation on a separate disk. Documentation runs to 11 pages giving you all you need to know about using Page Editor and its utilities. This program won't allow you to create your own version of new screen font but will give anyone with an SL or SD and a printer the opportunity to create interesting page layouts for all sorts of applications.

See disk set - price also list

DON'T FORGET FUTURA

D8#72 - FUTURA 1

D8#73 - FUTURA 2

D8#76 - FUTURA 3

D8#79 - FUTURA 4

D8#87 - FUTURA 5

D8#89 - FUTURA 6

D8#137 - FUTURA 7

D8#138 - FUTURA 8

D8#139 - FUTURA 9

D8#140 - FUTURA 10

D8#141 - FUTURA 11

D8#142 - FUTURA 13

D8#143 - FUTURA 13

D8#144 - FUTURA 14

D8#145 - FUTURA 15

D8#146 - FUTURA 16

D8#147 - FUTURA 17*

* double disk issue of \$2.80

D8#148 - FUTURA 18

D8#149 - FUTURA 19*

* double disk issue of \$2.80

D8#150 - FUTURA 20

D8#151 - FUTURA 21

RECENT ADDITIONS

D8#133 - JOYTRIDE

A great computer demo

D8#134 - BOSTERM

The top course program

D8#136 - ATARI CAD

A superb design program,

especially for those who

use circuit diagrams - our

best seller last issue

BUY NOW!

BARGAIN CASSETTES

Your choice of
any 5 cassettes for £1.50 plus 90p p&p
any 10 cassettes for £2.00 plus £1.20 p&p

180	✓ LOS ANGELES DRAT	✓ REVENGE II	✓
BOMB FUSION	✓ MASTER CHESS	✓ ROCKFORD	✓
DISPATCH RIDER	✓ MILK RACE	✓ SIDEWINDER II	
FEUD	✓ MR DIG	✓ SPEED HAWK	
FOOTBALL MANAGER	✓ NINJA	✓ SPEED ZONE	✓
GHOSTBUSTERS	✓ ON CUE	✓ STAR RAIDERS	✓
GUN LAW	✓ PANTHER	✓ TAIL OF BETA LYRAE	✓
HENRY'S HOUSE	✓ PENGON	✓ TWILIGHT WORLD	✓
INVASION	✓ PLASTRON	✓ UNIVERSAL HERO	✓
KIKSTART	✓ ✓	<i>TRANSDISK IV shows you how to transfer these to disk!</i>	

COMMERCIAL SOFTWARE STILL AVAILABLE VERY LIMITED NUMBERS

(Prices inc. p&p)

NIBBLER	DRUID	JUGGLE'S HOUSE
Disk £1.20	Disk £1.70	Cassette £1.70
MAXWELL'S DEMON	LANCELOT	BATALION COMMANDER
Disk £1.20	Cassette £1.20	Cassette £1.70

ORDER ITEMS FROM THE ACCESSORY SHOP WITH THE ORDER FORM
ENCLOSED WITH THIS ISSUE OR WRITE TO

PAGE 6, P.O. BOX 36, STAFFORD, ST16 1DR

TELEPHONE ORDERS ACCEPTED ON 0759 241181 USING ACCESS OR VISA

Features and OPINIONS

SPARTADOS

An overview and list of commands

compiled by
M. Tomlin

Spartados is probably the most sophisticated disk operating system developed for the Atari 8-bit system, and it boasts quite a list of features. It supports multiple sub-directories, knows no letters on the DTI, full time/date stamping of files, full random access to any byte within a file, batch command files, hard disk access, and a whole host of more specialist features.

Spartados keeps most of its code locked away under the operating system so you get more free memory for your programs. Even with the most powerful configuration, you will still have 33,541 free bytes available in Basic, and the smaller versions can have as much as 35,170 available. These figures should be set against 24,274 with Dos 2.0 and 37,502 with no Dos at all.

GET A CLOCK!

If you can find one (are or second hand) on IC's 8-Tape II battery backed-up clock cartridge and the correct time and date will always be available to be stamped on each new file created. The clock module plugs into the standard cartridge port and has a replacement socket on the top for another cartridge. It has been tried with Ardour, Mac/80, Penta/80, Atari/Arca, Atari Assembler/Editor and so on - and it seems completely transparent to everything. There is even a 25 character which allows you to access the clock directly from Basic.

US DOUBLER

Where Spartados really comes into its own is in conjunction with another IC's product - the US Doubler. This used to come in the form of two plug-in ICs for the 1850 drive which not only give it the ability to work in true double density but also speed up the data transfer rate considerably. Once these chips are installed, your drive is supercharged. Upon loading your Spartados master disk the old floppy, floppy control is replaced by a burst of rapid reading speed. Reading data from a disk is roughly four times faster than with a standard 1850 drive, but after allocating the work time the time taken to move the head around the disk (the average speed increase is nearly three times). The increase in speed of writing is less at roughly twice as fast. The other major feature of the US Doubler is to give access to true double density, giving 256K per disk as opposed to 90K in single or 120K in enhanced density. Double density actually uses 720 sectors per disk, as with single density, but each sector contains 256 bytes of data rather than 128. This does mean that you'll have to remember to format and write in standard mode if you wish to give a disk to somebody without a US Doubler.

SpartaDos Commands (vers. 3.2) and comparison with DOS 2.5

Dos 2.5 SpartaDos	Function
A DIRS	Disk directory, any drive (Dos 2.x format)
B CAR	Go to cartridge (if present)
C COPY/SCOPY	Copy file(s) (multiple drives)
D ERASE	Delete file(s) from disk
E RENAME	Rename file(s) on disk
F PROTECT	Protect file(s) from accidental erasure
G UNPROTECT	Remove erasure protection from file(s)
H n/s	Write Dos file (disabled during format by XINT)
I XINT	Formats disk (see also AINT)
J DUPEK	Duplicate whole disk
K SAVE	Save binary file (see also APPEND)
L LOAD	Load binary file (see also OFF LOAD)
M RUN	Run machine code at given address
N n/s	Create HIGHRAW (dispartion to always to memory)
O SCOPY	Copy file(s) (single drive) (see also B/COPY)
P AINT	Format (single density) Dos 2.0 model
APPEND	Save binary file at end of existing file
ATTATCH	Select batch file to run when float is present
BASIC ON/OFF	Turns internal basic on or off
BOOT	Set floppies to load when no Dos present on disk
STPARM	Modify hard disk drive access number
CHDIR	Give current disk statistics
CHD	Change time/date stamp on file(s)
CHVOL	Change volume name of disk
CHDIR	Makes new sub-directory
CWD	Change default path details for current drive
DATE	Set system date
DELDIR	Delete sub-directory (must be empty)
DIR	Shows path to specified sub-directory
DIR	Disk directory (Extended format: time/date/system)

DUMP	Print file as Ascii + hex digits to screen
SET ON/OFF	Type-ahead handler on or off
LOCK/UNLOCK	Protect whole disk from write operations
REDUMP	As for DUMP, but to print memory contents
RECK	Show current Dos format/headers/subsets
RESD	Load menu system (may be set as default)
OFF-LOAD	Binary load file, with effect
PAUSE	Wait for a key to be pressed (in batch files)
PORT	Change the RS-232 configuration
PRINT	Echo screen output to another device like F: or C:
PUTROM	Add rom address to binary file
RD	Set up ramdisk (many configurations available)
RPM	Test disk drive rotation speed
RS232	Load RS-232 driver for 850 module/PIC connection
TIME	Set system time
TDLINE	Load time/date handler line routine
TD ON/OFF	Turn time/date line on or off (requires TDLINE)
TRER	Shows all sub-directories/files (alphabetical)
TYPE	Shows Ascii file contents on screen
VERIFY ON/OFF	Turns disk write on or off
WOW	Turn off I/O redirection (disable PRINT/batch mode)
ZRAND	Address Z: for time/date handling from basic
-Basename	Kerning batch command file (parameter = .BAT)
Basename	Kerning machine code file (parameter = .COM)
Dir	Select new default drive number, n

but this is very simple since SpartaDos is intelligent enough to detect what type of disk it is using.

date stamping of files and the sub-directory facilities, and their ease of use from basic and other languages. SpartaDos can also read, write and format Dos 2 disks automatically, so you need never worry about which type you currently have in the drive. It can even handle a loaded disk if you happen to have one. If you can get hold of a copy (or even a photo copy) of the excellent manual all the better but as most people nowadays will only be able to get a copy without the manual I have included a list of most of the available commands which should help.

OVERVIEW

SpartaDos has so many features that it is impossible to describe them all in full, but the list of SpartaDos commands will give you some idea of its amazing power and flexibility when compared with Dos 2.5. I like the time-



JOURNEY INTO CYBERSPACE

*John S Davison
continues his
exploration of
the Internet*

PRACTICAL USE OF THE INTERNET

You can't have failed to notice that the Internet is now beginning to reach our everyday lives. It's not just mentioned in the mass media any more - they're actually begining to embrace it and already encourage its use. For instance, many radio DJs now regularly write communication from listeners via e-mail while they're on the air; an increasing number of TV programmes now have their own World Wide Web addresses where additional information can be found, and many adverts on TV, in magazines and newspapers, and even on poster boardings now quote the companies' WWW addresses along with their promotional messages.

But, there are still many people who view the Internet as a high technology toy used only by computer geeks who play with it, "because it's there" or who use it purely for Malicious forms of entertainment. This view generally stems from the fact that "non-believers" have no obvious practical use for it themselves. Such people need to use it "for real", i.e. with a specific purpose in mind, before they'll accept it for what it is - an incredibly rich source of information, entertainment, and increasingly so for commercial services (from variously so for "e-business", "e-commerce", or even e-shopping). So, over the next few issues I'll be looking at examples of practical use of the Internet to see how practical they really are.



OK, I know this isn't *Start* specific, but there's no reason a suitably configured *Start* couldn't be used in the ways described here.

TO AMERICA VIA CYBERSPACE

The practical use of the Internet was recently highlighted to me as a family following my older son's decision to take up a job offer in the USA. Long time readers of *NAU* may remember John Jay - he wrote over 150 pieces for the magazine several years back while he was still at school. And thanks to this early experience (plus a lot of hard work) he's subsequently been able to build himself a very successful career in the magazine publishing industry. As we'll see, the Internet has figured prominently in his latest career move.

He recently accepted the post of Editor in Chief of *Electronic Gaming Monthly*, a highly popular magazine for users of games consoles such as the Nintendo 64, Sony PlayStation,

and similar. The magazine is based near Chicago and much of the communication between John and his new employer was conducted via Internet e-mail, while he was still working in the UK. The time difference of 5 hours between London and Chicago makes telephone communication inconvenient, and e-mail provided the perfect solution. So, the Internet proved a real benefit to John, right from the start of negotiations.

FINDING A HOME

Once he'd accepted the job the hard work begins. It's hardly enough moving to a new job in a different part of the UK, but when it's 5,000 miles away on a different continent the problems increase tenfold at least! The first challenge was to find somewhere to live. But how do you do that without spending weeks living in a hotel while trying to househunt and simultaneously attempting to get stuck into the new job? The answer - use the Internet as a househunting tool before you move to the new job.

John used a search engine to find the Web sites of companies that advertised properties for rent. The one he eventually settled on was called *Real.Net* (<http://www.real.net>), if you care to take a



Floorplan of a property on Real.Net

Picture of property development in Wheaton on Real.Net



look. This lets you search for rental properties available in America (all 50 states), Canada, and in certain other parts of the world too. You tell it the type of property you want, e.g. an unfurnished apartment, and it then steps you through a series of pages where you refine your requirements. You tell it which USA state you want to live in (Illinois is John's choice), and then which major city area (Chicago). It then presents you with a list of Chicago suburban areas. After selecting the suburbs of interest (Gardner and Wheaton in John's case, as his new company is based there) you tell it the size of property required (e.g. one bedroom, two bedrooms, etc.) and the price range to look for. A list of available properties to suit your requirements and pocket then appears. Clicking on these brings up further information, including phone, fax, or e-mail details of the listing agents and, most usefully, photographs and floor plans of each property, so you can get a good idea of what they're like before going to view them for real.

If there are lots of properties meeting your criteria you can produce a prioritized list based on attributes available. For instance, you may want somewhere that allows pets (important to John as his two cats emigrated with him), or that has air conditioning (strictly a luxury at the moment) and microwave included. Properties meeting these requirements are then placed at the top of the list, so you don't have to search every house from yourself. Why or what?

RESEARCHING THE LOCALITY

Obviously, you also want to know something about the area in which the available properties are located. These are John's choices were in Wheaton, so by using a search engine that references to Wheaton you can then click your way to the City of Wheaton home page (<http://city.wheaton.il.us/>) instead. Through this and other links contained there you can find out virtually everything you need to know about Wheaton and what's currently going on there. It's really a good tool.

John and his wife Ali (yes, he's married, so then fire out to Chicago to view the apartments advertised from first list). They were on the point of signing up for one of them when the listing agent showed them a brand new development in Wheaton that was only just about to be put on the Real.Net Web site. They found a town house in this complex was even better than their original choice, they went for this instead.

Real.Net also offers other facilities associated with moving house. For instance, there are details of self storage units you can rent to store your belongings in while you sort a new accommodation. There are links to companies from which you can hire self-drive removal vans for do-it-yourself removals. There are details of local

schools and child care services, cable TV hook-up companies, and insurance companies. If you don't have any furniture you can even rent some from here too. John and Ali's new furniture was being shipped from the UK by sea and took about six weeks to reach Chicago. They'd needed the town house well before the furniture delivery date, so faced the prospect of a month or more without a stick of furniture in the place. The solution was to take out a short term rental on a handful of furniture and household appliances until their own stuff arrived.

You can also find the location of the properties using an on-line map. There are maps on Real.Net showing Chicago and surrounding areas, but there's a much better map facility available through Yahoo, at <http://map.yahoo.com>. You key in the property's full postal address and a fully zoomable map of the area appears, with your selected address marked on it. You can zoom out until the map shows the whole USA or zoom in until you have a local streetmap of the immediate area around the specified address. If you then supply the address of your current location the system will provide driving directions from there to the property you want to view. It is rather like the AutoRoute program available for the Acet 3T a few years ago.

E-FLOWERS

When John moved to Chicago we've kept in touch with him on a daily basis through e-mail. Phone calls cost far more than e-mail.



City of Wheaton Home Page



Map of Wheaton area by Yahoo Maps

so are reserved for weekends and special occasions. One such occasion was Mother's Day, and as well as placing his own John also sent her a large bouquet of flowers. He selected, ordered, and paid for these via the Interflora Web site (<http://www.interflora.co.uk>). Here you can choose a price range and style of bouquet you want to send, specifying the types or sizes the colours of flowers to be included. Photographs of suitable bou-



ports are displayed, from which you choose the one you want. You then enter delivery address, time and date for delivery, and a message for the accompanying card (plus your credit card details, of course, and InterPort take it from there. It works amazingly well. His main was so impressed that she used the same service shortly afterwards to send flowers to HER man, to cheer her up following an illness. However, she will be nagging about sending credit card details over the Internet, so she placed our local InterPort kiosk instead to order the item she'd chosen on the Web site. Amazingly, it cost more to do it over the phone than via the Internet.

Well, I think the above is proof positive that

the Internet does have practical applications. Admittedly it would have been done in other ways, but would have taken far longer to achieve. Introducing to one of the Internet's biggest plus points, and it certainly helped John and Jill to rapidly set up their new life in America.

Several months on from John's story we decided to take a holiday in the USA. Naturally, we wanted to go to Chicago to see Jim, but we also decided to visit friends in Texas whom we'd not seen for several years. You can probably guess what's coming next. Yes, we arranged most of the trip with the help of the Internet. It's a whole story in its own right, so we'll be looking at this in the next issue.

NAU Internet Contact List

The following NAU readers would welcome a e-mail contact from other Atari users. If you'd like to be added to this list please drop an e-mail note to John S Denton at the address below.

Daniel Bevanstock

Paul Bramley

Paul Carlson

Johnny Chan

Kevin Cooke

Michael Current

John S Denton

Damian Dixon

Gary Dundas

Derek Fern

Dean Gersmarty

Joel Goodwin

Paul Harfoot

Gordon Hopper

Fred Moller

Ann O'Driscoll

Allan Palmer

Paul Rison

Paulo A Rodrigues

Brad Rogers

Nigel Turton

Hanning Wright

Daniel Yelland

Bryan Zilwood

dbevanstock@rmitral.co.uk

p.bramley@students.qut.edu.au

paul.carlson@fn.usa

johnchan@ptera.net

kj.cooke@isp.usps.an.au

mcurrent@colorado.edu

john_denton@compuserve.com

damian@netnet.co.uk

davedundas@hotkey.net.au

101750.2440@compuserve.com

dgs@olera.net

jgoodwin@villco.co.uk

143378.97@compuserve.au

us2556@freemut.victoria.bc.ca

fmoller@dxr.nl

annod@iol.ie

100844.1040@compuserve.com

risonp.net@nhs.uk

roop24454@mail.telapco.pt

brad@planetusa.denton.co.uk

optatari@msn.com

huff@holgomet.se

yh182434@emul.staffs.ac.uk

b.zilwood@exeter.ac.uk

contact ... contact ... contact ...

Only one notice for CONTACT this issue so Gary Partington gets a huge free advert

COMPUTER HARDWARE FOR SALE

Atari 800XL computer (Dual 0/5), 1020 printer, 1027 printer, 1029 printer - all in good working order with power supplies and manuals

- OFFERS INVITED

Modem cable - RS232 25 pin (F)/9pin (F) to 36 pin (M) - £5

RS232 'D' Connector Converter 25 pin (F) to 9 pin (M) - £3

Printer Cable - Centronics - 32 pin (F) to 25 pin (M) - £5

5.25 inch computer disks - (50) (home sew) - £10

5.25 inch disk drive cleaning disk kit - £5

Tel. Gary Partington on 01788 613624

FREE TO SUBSCRIBERS

The CONTACT notices in free of charge to subscribers who wish to sell their equipment or contact other readers. Space is limited so we request that entries be kept as short as possible. Excessively long entries may be heavily edited or ignored. Send your CONTACT notice on a separate sheet of paper (not as part of a letter) to CONTACT, PAGE 8 PUBLISHING, P.O. BOX 94, STAFFORD, ST18 1LR

FOR SALE ... WANTED ... PEN PALS ... ADVICE ... HELP

PROGRAM LISTINGS

Certain program listings which are too long to include in the magazine may be obtained free of charge as printed listings in type in ALL programs are, however, included on the Issue Disk which is available with each issue. Remember this disk also includes BONUS PROGRAMS which do not appear in the magazine. If you would like the type in listings please write or telephone indicating which listings you require. Please note that there are not necessarily extra listings for every magazine.

Write to LISTINGS, NEW ATARI USER, P.O. BOX 54, STAFFORD, ST16 1DR or telephone 01785 241153