

WORKBENCH

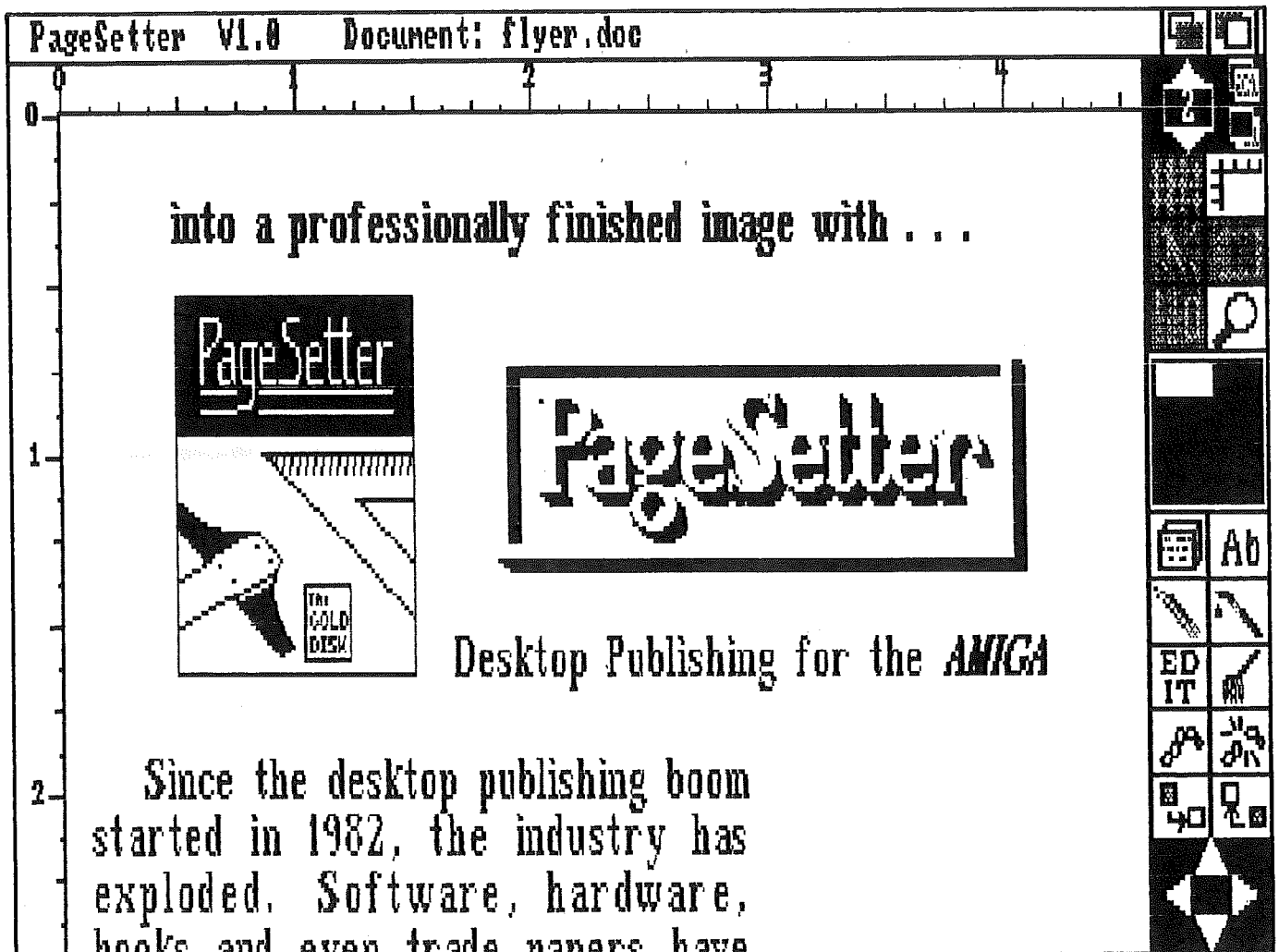
FOR THE COMMODORE AMIGA USER

Registered by Australia Post – Publication No. VBG7930

Number 10

Circulation: 500

March 1987



Next Meeting

Sunday, March 8th, 1987 at 2pm

AUG meetings are held at Victoria College, Burwood Campus
in Lecture Theatre 2. Melways map 61 reference B5.

Amiga Users Group, PO Box 48, Boronia, 3155, Victoria, Australia

AMIGA is a trademark of Commodore—Amiga, Inc
The Amiga User's Group has no affiliation with Commodore

AMIGA™ Users Group

P.O. Box 48, Boronia, 3155, Victoria, Australia

Amiga Users Group

The **Amiga Users Group** is a non-profit, self-help group, made up of people interested in the Amiga computer and related topics.

Club Meetings

Club meetings are held at 2pm on the second Sunday of each month at Victoria College, Burwood Campus, in Lecture Theatre 2. Details on how to get there are on the back cover of this newsletter. The dates of the next few meetings are:

Sunday, March 8th at 2pm
 Sunday, April 12th at 2pm
 Sunday, May 10th at 2pm (Mother's Day)

Production Credits

This month's **Amiga Workbench** was edited by Peter Jetson. Equipment and software used was: TurboDOS S-100 computer, Diablo 630 printer, Gemini 10x printer, Wordstar and Fancy Font.

Copyright and Reprint Privileges

Articles herein that are copyrighted by individual authors or otherwise explicitly marked as having restricted reproduction rights may not be reprinted or copied without written permission from the **Amiga Users Group** or the authors. All other articles may be reprinted for any non-commercial purpose if accompanied by a credit line including the original author's name and the words "Reprinted from **Amiga Workbench**, newsletter of the **Amiga Users Group**, PO Box 48, Boronia, 3155".

Contributions

Articles, papers, letters, drawings and cartoons are actively sought for publication in **Amiga Workbench**. It would be appreciated if contributions were submitted on disk, since that means they don't have to be re-typed! We have access to a wide range of computers, so we should be able to accept almost any type of disk, but Amiga disks are certainly the easiest. All disks will be returned! Please save your article in **text-only** format. Absolute deadline for articles is 14 days before the meeting date. Contributions can be sent to:

The Editor, AUG, PO Box 48, Boronia, 3155
 (Note the new address)

AUG Users Group Disks

Disks from the **Amiga Users Group Library** are available on quality 3.5" disks for \$10 each including postage on AUG supplied disks, or \$2 each on your own disks. We can also provide 80 track 5.25" Amiga format to special order. Please enquire. The group currently holds 66 public domain volumes, mostly sourced from the USA, with more on the way each month.

Member's Discounts

The **Amiga Users Group** is currently negotiating discounts for its members on hardware, software and books. Members will be notified when negotiations are complete.

Currently, **Technical Books** in Swanston Street in the city offers **AUG** members a 10% discount on computer related books, as does **McGills** in Elizabeth Street. Just show your membership card. Although we have no formal arrangements with other companies yet, most seem willing to offer a discount to **AUG** members. It always pays to ask!

Membership and Subscriptions

Membership of the **Amiga Users Group** is available for an annual fee of \$20. To become a member of **AUG**, fill in the membership form in this issue (or a photocopy of it), and send it with a cheque for \$20 to:

Amiga Users Group, PO Box 48, Boronia, 3155

Amiga Users Group Committee

John Holland . . . (Co-ordinator) . . .	348 1358 Carlton
Bob Scarfe . . . (Vice Co-ordinator) . . .	376 4143 Kensington
Ron Wail . . . (Meeting chairman) . . .	878 8428 Blackburn
Eric Salter . . . (Secretary) . . .	861 9117 Kew
Paul Radford . . . (Treasurer) . . .	663 3951 BH only
Neil Murray . . . (Membership) . . .	792 9666 Dandenong
Bohdan Ferens . . (Purchasing) . . .	792 1138 Dandenong
Geoff Wood . . . (Book Librarian) . . .	580 7463 Aspendale
Geoff Sheil . . . (Software Librarian) . . .	509 3151 Armadale
Ron Wail . . . (Software Librarian) . . .	878 8428 Blackburn
Peter Jetson . . . (Newsletter Editor) . . .	762 1386 Boronia
Stephen Thomas	830 5783 Canterbury
Jo Santamaria	836 9129 Canterbury
Ron Van Schyndel	882 7264 Hawthorn
Mike Creek	878 9039 Blackburn

When phoning committee members, please try to be a bit considerate and not call at meal-times, late at night, or during popular TV programs. If you only have a general query, try to ring the member who lives closest to you.

ELECTRONIC ARTS - DELUXE PAINT II - Wow!!!

At a recommended retail price of \$249.00, this new updated version has to be the best value for money of any graphics related program on the market today.

As a constant user of **DPaint** for television and video work, (titles, logos, graphics etc.), I have been eagerly awaiting this update. A reported **fifty** new additions have been added to the **DPaint II** program, and it shows!

DPaint II comes packaged in a presentation box, similar to **Aegis Animator/Images**. The enclosed 150 page manual is not an addendum to the original instruction book, but a completely new version that caters for both the beginner and seasoned user. And it is spiral bound, thus avoiding all those 'spine cracking' problems found with the **Aegis** books.

DPaint II's built-in **Workbench** is 1.2 version 33.44, and thus requires **Kickstart 1.2** to boot up. The 'Keydisk' system is used as copy protection. After booting your backup copy of the program disk you are asked to insert the original program disk to confirm its key number. Once verified, your backup is loaded.

Opening **DPaint** brings up a screen requestor "Screen Format". You have the option to "Load All" the program into ram, thus using up 300k of memory, (100k to **Intuition** and 200k for the program). One **Lo-Res** picture will eat another 40k, and if you're using a second disk drive discount 20k, only leaving you around 150k to play with. If you "Load All" and then access **Hi-Res** you will only have four colors available. Operating speed is the only advantage of loading the program into ram. The second option is "Swap", where only the **DPaint** work screen information is loaded into ram. This leaves a lot more memory available for all those complex paintings, but the program disk has to be accessed after selecting menu commands.

Then, you can select one of the four screen resolutions:

LO-RES	320 x 200	32 colors
MED-RES	640 x 200	16 colors
INTERLACE	320 x 400	32 colors
HI-RES	640 x 400	8 colors

(Assuming you are running 512k. More memory = more colors.) Press 'OK' and you're on the selected work screen.

DPaint II is a two-disk package, the second "**Art Disk**" has twenty nine **Lo-Res** paintings, one **Med-Res** example and six sample brushes. These relate to the manual tutorials and show off various aspects of the new additions. One painting, 'Lost Ark' (Raiders Of The Lost Ark), demonstrates just what can be done with the cycle mode - it's a real eye popper!

Deluxe Paint II Menus

Picture Menu

Accessing external drives or hard disk has been sped up (compared to the original **Deluxe Paint**) by including **df0:**, **df1:** and **hd:** 'click' boxes.

The **Delete Picture** requester now takes care of the problem of getting rid of unwanted files without exiting the program. The **Print** requester allows for **normal** or **sideways** orientation, **Black and White**, **Grey Scale** or **Color** printing, **Left** and **Right** margin setting, **Paper Height** and **Width**

percentage, and **Number of Copies**. When printing is underway, a **Stop** click box is shown on the screen, thus allowing printing to be stopped at any time.

Another requester, **Color Control**, includes the sub-menu **Palette**, now with four cycle settings plus a cycle direction arrow. Three sub-menus in color control are **BG -> FG** to change the background color to the foreground, **BG <-> FG** which does it both ways, and **Remap** which allows a painting created with a different color palette to be matched to the current screen palette.

The **Spare** menu includes **Swap** to change to the second screen and **Copy to Spare** for copying the current screen to screen two. **Merge in Front** shifts the spare page in front of the current screen, while **Merge in Back** places the spare page behind the current screen. **Delete This Page** deletes the page that is currently showing, and re-allocates the memory for other uses.

The **Page Size** requester deals with the size of the picture area on the screen and allows you to customize the **Width** and **Height** of the screen display. **DPaint II** will recognize page sizes up 1008 x 1024 pixels, but you will have to use the arrow keys to scroll around the entire picture. (In **Lo-Res** with 512k memory and a screen size of 1008 x 1024 you can only have two colors). If you don't want to set your own page size you can select one of three others: **Standard** which is 320 x 200, **Full Page** (320 x 340) which matches an A4 sheet of paper, or **Full Video** (352 x 226) giving you a full screen image of your work for dumping to videotape.

PAL users, (that's most of us) please note:- Using **DPaint II** with **Workbench 1.2** still does not give you a full screen image. The bottom quarter of the screen is still unavailable as work area regardless of what page size you set. The Yanks are a pig-headed lot. Until they come to realise that their 525 line television system, (N.T.S.C), is the world's most inferior system, us **PAL** (625 line) users won't get decent software. A Sydney-based graphics company has so impressed the Americans with its **REAL** video-graphics software that it has just done a deal with one of America's largest computer graphic production houses to use the Aussie software).

Show Page displays your current page in its entirety, but in a reduced format, while **Screen Format** brings up the screen format requester with two options, **Same Page Size** or **New Screen Size**, allowing you to resize the picture for the resolution you are about to enter.

Brush Menu

The only real change here from **DPaint I** is **Handle**, which lets you specify how you hold a brush, in the **Centre** or **Corner**. The latter allows the brush to be held by any of the four corners.

Mode Menu

Matte uses a custom brush in its original form and the areas of the brush that match the picture background color are transparent. **Color** uses the shape of the brush and fills it with the current foreground color. **Replc** uses the brush in its original form, except that no colors are transparent, while **Smear** does exactly that, giving a watercolor effect. **Shade** creates subtle shading effects on colors in the picture that are included in the current cycle range, and **Blend** uses additional shades from the color palette to blend one color with the next. **Cycle** uses the current brush and cycles through all the colors in the current cycle range as you

Have you noticed our change of address?

text into this screen. Dumping a screen's RastPort simply involves copying some information from the screen's Screen structure into the printer request block and sending the request to the printer driver.

Using The Narrator

Just for fun, I decided to try to get the program to speak the phonetic version of the Thai words. I did not really expect to get very good results (considering the quality of the Amiga's English pronunciation) but it was fun trying. There were two main problems I encountered and only partially overcame.

Unfortunately, Thai has got a lot more vowel sounds than English. The narrator device simply did not know how to say all the sounds I needed. The best I could do was either to pick the closest sounding one or string a few together. As I am not really sure myself what they are meant to sound like, I cannot be certain of the results.

Thai is a tonal language which means that the pitch of the sounds also affects their meaning. In Thai there are 5 basic 'tones': common, low, high, rising and falling. In English, when asking a yes/no question for example, we normally raise the voice at the end of the question ('Are you well?'). This must not be done in Thai. Just to make things that little bit more interesting (or confusing), the duration of sounds is also important (short or long). Each syllable in a Thai word then has its own sound (usually consisting of an initial consonant, a middle vowel and an optional final consonant), a tone, and a duration. Trying to get the narrator to speak a multi-syllable word can get quite tricky!

To be able to get total control per syllable, it was necessary to send the narrator device one syllable at a time. This unfortunately resulted in significant pauses between each syllable as the narrator thought it was the end of a sentence. I have found no way to get around this problem. Sending a separate command per syllable makes the duration easy to control. The common, high and low tones are also simple - just use the robotic (monotone) voice and change the pitch. The rising and falling tones are harder, but are possible by using the natural (contoured) voice. Adding a '.' at the end of the syllable (to make it think that its the end of a sentence) generates a falling tone and adding a '?' generates a rising tone.

Conclusions

The Amiga with Intuition sure makes it easier to do nice interfaces. It provides lots of easy, ready to use functions to handle gadgets and menus. What I really would like though, is an easier way of designing the menus, gadgets, windows etc. without having to recompile the program. I have heard (in other words I really don't know) that the Mac allows such definition files that can be modified to change the interface to the program without the program requiring recompilation. Looking at the latest public domain disks, however, there seems to be a gadget editor (egad) which I will have to have examine, and there was a menu editor on an earlier disk, too.

The next things I want to have a go at are the animation routines and a bit of music. Then I can get onto some serious stuff, like writing a few games!

-- Alan Kent

Sidecar - A User's View

I received my Sidecar two days before Christmas, it was the last bit of my system to arrive. I had, like everyone else that had one on order, waited and waited for its arrival. I'm very happy with it for several reasons. I now have a system that has the fantastic graphics, sound capabilities and processor speed of the Amiga yet runs plain IBM software as well.

You may be wondering why anyone would want to use an Amiga as an IBM clone. Well there are several good reasons. The main benefit is that a Sidecar + 20 Mb hard disk drive (IBM type) combination is about \$1,000 cheaper than getting a dedicated Amiga hard disk drive, yet you get another whole computer thrown in! The Sidecar also doubles as an expansion box for 2Mb of Amiga RAM, it also gives you a 5.25 inch disk drive and access to any IBM addons you put in the Sidecar's slots like clock-calendar cards and co-processor chips. Least of all it gives you access to IBM software. This last point may seem of little significance but when file transfer software is available shortly, porting old databases that have taken years to develop or spreadsheet formulae that take two hours to read let alone type out again can be used by the Amiga.

Software allows the Amiga and Sidecar to communicate using the interrupt driven interface. Under special circumstances either machine can use the facilities of the other. This allows an extended multiprocessor environment using both the 68000 and the 8088. The interface area is 128K of dual-ported memory which directly attaches to the Amiga's expansion bus. There is no degradation of performance on either machine.

As the video portion of the PC memory is directly mapped into the dual ported RAM, any unusual video effects using the ROM BIOS are supported. Both MONO and COLOUR modes can be emulated at the same time.

Any PC compatible hard disk drive can be placed in one of the three IBM compatible slots. The disk can be partitioned to give access to both the Sidecar and to the Amiga. As PC-DOS allows you to partition a hard disk drive, you can set aside 10Mb for MS-DOS and 10Mb for AMIGA-DOS. I decided not to get a second 3.5 inch drive and to rely on the hard disk drive instead. The speed of the hard drive is great! It is envisaged that expansion memory can be fitted in the Sidecar, possibly 2Mb.

The Sidecar runs the 8088 at 4.77 Mhz. 256K comes as standard, however 640K can be placed in one of the slots or a maximum of 512K can be placed straight onto the motherboard. The disk is standard 360Kb in the normal 2 sided format on a 5.25 inch disk. A 3.5 inch drive can be plugged into the external connector and this can be formatted to 720 Kb capacity. The 3.5 inch drive will allow access to IBM programs distributed on this size media.

It is possible to have more than one IBM window open at a time, so that although you can not multitask, at least you have some of the windowing benefits without having to use additional IBM operating systems. Of course you can use the multitasking Amiga to run as many programs as you like while the Sidecar runs its application. So while a large IBM database is sorting you can click over to an Amiga painting, or do some wordprocessing. The machines, although closely integrated, can be reset individually without affecting the program running in the other or you can reset both simultaneously.

The procedure for booting the Sidecar-Amiga combination is simple. Place the Kickstart disk in the Amiga drive and turn on the power, the Sidecar and Amiga will power up and the familiar Workbench hand will appear. Next, insert the supplied Sidecar startup disk. This will set up a normal workbench screen. The Sidecar is now ready to use at the click of its icon.

I received the Sidecar box with some 5.25 inch and 3.5 inch disks, a DOS manual and a set of installation instructions. Everything made sense! Hooking up the Sidecar was a simple matter of removing the 'bus' cover on the disk drive side of the amiga, lining up the two units and pressing gently together until they were flush. The front of the Sidecar contains two mouse ports and a 5.25 inch disk drive. Power for the Amiga is supplied from the Sidecar (via a short cord that neatly reaches to the Amiga socket) and the original Amiga power cord is now used as the sidecar's cord. The Amiga's power switch is left on so the Amiga/Sidecar is powered up by the Sidecar switch only.

Before I had done all this attaching, I had installed the 20Mb hard drive in the Sidecar. I have a mountain hard card type which takes up one and a half of the sidecar slots (the half slot can still have a half length card installed in it). The installation was easier than I had imagined, it was a matter of simply removing the cover screws on the Sidecar, sliding the cover off and firmly seating the drive in the slot.

I now had to set up the works which was also pretty simple as each step of the way was explained in the installation notes. The 3.5 inch disk is booted (after kickstarting) and the Sidecar comes to life. If I had only the Sidecar, that would be all I would have done. The Sidecar would be opened by clicking on its icon, selecting the colour or monochrome drawer and waiting a few seconds as the IBM window is drawn. In my case I had a little more to do to set up the hard drive. The procedure was also explained in the notes and involved the use of a couple of programs. The first set up the DOS partition then another formatted the DOS section, the next set up the AMIGA partition and the last formatted the AMIGA section. The whole process took about 10 minutes and only needs to be done once.

I followed the remaining instructions and made a hard disk drive boot disk. This disk acts just like a Workbench disk but it automatically sets up the Sidecar and assigns the harddrive as the default drive which I have named AM:. The AM: disk icon appears where you would normally see the workbench disk icon. Now when I click on the AM: icon, a window opens and drawers appear on the screen in a flash. Amongst the drawers is the sidecar one which when clicked on leads to IBM. The hard drive being the default contains all the Amiga commands, preference information etc. putting an end to disk swapping on my one floppy system.

The IBM screen opens within the normal Amiga window. The borders of the window can be removed to give the appearance of a completely normal IBM screen on the monitor. Colours, cursor flashing speed, window size, etc can be controlled by using the menu selector. This gives fine control over things that would normally not be available to the IBM user.

The Sidecar uses the 8088 chip running at standard IBM speed (4.77 Mhz). It is a completely separate machine that can be reset independently of the Amiga. I have not found any software that does not run as expected, even programs that use strange screen updating methods and unusual keyboard reading routines run as normal.

Currently the parallel port on the Amiga is used by the Sidecar so that only one cable and one printer is required. The serial port is not accessible from the Sidecar yet. File interchange between IBM and AMIGA formats are also on the way.

The only problem I have had with the system so far is that because the hardcard does not have an "active" light I have turned the machine off when the drive was running. This has the delightful effect of making your disk appear unaccessible the next time you boot! When you next boot, you get a system warning that the hard disk is "not a DOS disk". All is not lost however, this is a safety feature of Amiga-DOS. When DOS is reading a disk, it sets a flag on track 0 of the disk to say that the disk is a non-DOS disk. When it has finished its activity and just before going back to the next task it resets the flag to say the disk is a DOS disk. If you power down before the flag is reset to normal then next time you boot you will get this not DOS message. It is simple to correct, just take a copy of the first track and place it in a file somewhere ready for recopying back should you do the inexcusable like I have done twice!

Not many people in Adelaide seem to have Sidecars, so I would certainly look forward to any correspondence about the subject, please write to me at this address:

Mike Simpson
49 Penang Ave.
Col. Light Gdns.
Adelaide, S.A., 5041

AMIGA Dealers:

Why not advertise in

Amiga Workbench

An advert this size costs only

\$20

(from camera-ready artwork)

Other sizes are available:

Half Page: **\$40**

Full Page: **\$70**

Double page: **\$120**

AMIGA Users Group
PO Box 48, Boronia, 3155

Public Domain Update

If you could only have **one** demo program for your Amiga, I think it would be the **Juggler**, on disk 47. Just the sort of program to make the owners of other computers drool! Also on the same disk in the latest upgrade to **vt100**, for those using this fine terminal program. At last, the bell works!

Fish disk #50 contains an upgrade to the public domain 68000 assembler, **ASM**. In the future, as soon as we can work out a few details, AUG may run a 68000 assembler course using this program. Also on the same disk is **breakout**, in 3-D. Get out your old pair of red/green cardboard glasses, the game is great! An update to **Wombat**, the terminal program, is also available on this disk.

On disk #53, you'll find some Aegis Animator **Animations**, and the first results of an interesting project to write public domain assembler versions of the standard Amiga-DOS commands. The stated purpose of this project is to "wipe out BCPL"!

Fish Disk #47

- 3D-Arm - The first stage of a larger project to study goal directed animation of articulated objects (objects which can move with various degrees of freedom but are constrained by attachment to other links within the object). This program demonstrates goal directed animation with a simulated 3-dimensional robot arm. Includes source.
- Juggler - Stunning animation of a robot juggler with ray traced reflective spheres. Uses HAM mode display and sound effects to boot!
- Vt100 - Version 2.4 of Dave's vt100 terminal emulator. Includes xmodem and kermit file transfer protocols.

Fish Disk #48

- Bru - Alpha 1 version of a hard disk and/or file archiver written for Unix. This is a vanilla port, with very little Amiga smarts, but it is usable. This is also the last freely redistributable version.
- Comm - A vt100 like terminal emulator with built in support for key functions and a phone number database. Version 1.30.
- Csh2.04 - Version 2.04 of Matt's csh-like shell. Includes source.
- Csh2.04M - Version 2.04 of Matt's csh-like shell, heavily modified for Manx C. Includes source.
- Diskperf - A disk benchmark program which runs on both Unix and the Amiga.
- Du - Computes disk usage of a file or directory (including subdirectories). Very handy.
- MemWatch - Memwatch is a program intended to sit in the background and watch for random trashing of low memory by an application under development. If it detects a write to low memory, it repairs it to what used to be there, then puts up a requester indicating what damage had been done. Very useful!
- Profiler - A realtime execution profiler for Manx 3.30E. Using this you can identify what sections of your program are using the most time, or being executed the most, and

optimize as appropriate. I was able to double the speed of some programs by using this tool. Includes source.

Fish Disk #49

- Cycloids - An updated version of the hypocycloids (electronic spirograph) program first released on disk number 27.
- DirUtil - An enhanced version of the dirutil from disk number 35. Dirutil is a program to wander around a directory tree and perform various operations on files. See README file for list of new features.
- Multidef - A program to scan a collection of object and library files and identify multiply defined symbols. This is useful to locate subtle problems where user defined symbols override library symbols.
- MyUpdate - A disk update utility that also provides support for automatically stripping comments from C header files and for interactive verification on a per file basis.
- Plot - Program to compute and plot 3 dimensional functions. Includes hi-res interlace mode for maximum detail.
- Polygon - A moire type pattern generator but with color cycling.
- QMouse - A program to query the status of the left mouse button and return a status code. The return code can be used as a WARN condition in a CLI execute file to alter execution. Also includes a separate program to toggle interlace on and off. Both are written in assembler and are very small.
- Touch - A simple command to set the date of a file to the current date. Uses new supported method of setting a file datestamp (rather than reading and writing back a byte).
- Trees - Another "tree growing" program, similar to the one on disk number 31, but more extensive.

Fish Disk #50

- Asm - A shareware macro assembler, submitted by the author. Asm is a 68010 macro assembler that is compatible with the assembler described in the AmigaDOS manual. This is version 1.1, an update to the version on disk number 46, and includes bug fixes, additional standard Motorola mnemonics and a simple startup code module.
- BreakOut - A 3-D game. Requires 3-D glasses.
- DiskZap - Program to "edit" a disk, sector by sector. Version 1.1.
- FirstSiliCon - A smart input line interpreter that provides a separate command window with full editing and recall of previous commands. Actual name is "Sili(Con:)".
- Missile - A nice "missile defense" game. Written 100% in assembler (source available from author) and is very fast!
- PerfectSound - PerfectSound demo, with a sound editor and sample sound files. Looks very nice.
- Sizzlers - Graphics demo programs.
- UnixArc - A version of arc suitable for Unix System V.
- Wombat - Yet another terminal emulation program. Version 3.01. Has user settable cursor,

programmable function keys, vt-102 emulation, auto-dialing, loadable settings files, and more.

Fish Disk #51

- Bison - A replacement for unix "yacc" command. This is from the GNU (GNU is Not Unix) effort, and was obtained from the Free Software Foundation. This is a working update to the version on disk number 4, provided by Johan Widen.
- Compress - This is an update to the compress version released on disk number 6. It is tested and seems to work well except for a couple glitches noticed at the last minute (see the README file). Compress reduces the size of the named files using adaptive Lempel-Ziv coding. The amount of compression obtained depends on the size of the input, the number of bits per code, and the distribution of common substrings. Typically, text such as source code or English is reduced by 50-60%. Compression is generally much better than that achieved by Huffman coding (as used in "pack") or adaptive Huffman coding (as used in "compact"), and takes less time to compute.
- Cos - A "Wheel Of Fortune" style game in AmigaBasic.
- DifSsed - Dif produces a difference file which lists the line by line differences between any two text files. These differences can be fed to ssed (a stream type editor), along with the original text file, to remotely reproduce the other (new) file. Binary only.
- Sq,Usq - Portable versions of the CPM sq and usq utilities.

Fish Disk #52

- Assign - Replacement for AmigaDOS assign command, written in C.
- Fractal - A fractal program which produces a random square fractal terrain.
- HAMpoly - A polygon drawing demo which uses HAM mode to produce polygons of any of the possible 4096 colors.
- MxGads - Demonstrates how to make mutual exclusion of boolean gadgets work, even if they contain GadgetText.
- Poly - A polygon drawing demo using the AreaMove, AreaDraw, and AreaEnd functions.
- Tek4010 - A Tek 4010 emulator.
- Vdraw1.16 - Shareware drawing program submitted by the author. This version uses menus but includes some items not in version 1.19, also on this disk.
- Vdraw1.19 - Latest version of Vdraw, with a completely new iconic user interface, and some other new features including a color palette mixer and DPaint style brushes, extended cutting features, area locking, and a flexible printing interface.

Fish Disk #53

- Animations - Some animations made with Aegis Animator. Includes a player program.
- ARCre - Creates rename scripts so that files with long names can be easily archived and

- Arp - Some preliminary programs from a program started on BIX by Charlie Heath to replace all the BCPL AmigaDOS programs with assembly equivalents. Includes "break", "cd", "chmod", "echo", "filenote", and "mkdir".
- Compiler - An optimizing C compiler for the 68000. Has successfully compiled itself on a 68000 based Unix system V machine, but has yet to be fully ported to the Amiga. It will currently produce assembly output for simple C routines on the Amiga, but needs lots of work to be really useful as anything other than a learning tool for budding compiler guru's.
- SpreadSheet - A simple "Visicalc-like" spread sheet calculator. Also known as "vc" but there is currently a debate about whether or not that name is a registered trademark. The version on disk number 36 was based on an earlier version of the same source. This release also includes source code.
- TarSplit - A port of the OS/9 program that extracts files from Unix tar archives.
- UUencode - Programs to encode/decode binary files for transmittal via mail or other text-only methods. The binary file is expanded by approximately 35% for transmittal. This release is an update to the version on disk number 38, and includes a simple line-by-line checksum technique that can read and write files compatible with the older uuencode/uudecode.

Reading The AmigaDOS 1.2 Documentation Files

At the last meeting, someone mentioned that they were having trouble reading the documentation files on the 1.2 update disks because they were in Infominder format and they did not have Infominder. When I collected my update kit on Friday, I found out exactly what they were talking about, typing the files resulted in instant hieroglyphics!

Luckily there are a couple of solutions to this problem, three if you consider buying Infominder which I didn't.

Firstly you can use the demo version of Infominder which is on the Extras disk to read the files. This works but doles out the information in small chunks and I haven't been able to get it to print anything on my printer yet.

The other option is to convert the Infominder file to a proper text file. This isn't as hard as it may seem at first glance, mostly it just involves stripping out a whole bunch of control characters and some command lines. Doing this manually would be a real pain, so the program in listing 1 was written on Saturday afternoon to do the job.

This program is written specially to process the 'allchanges.hbin' file on the Extras disk as it was the first one I wanted to read. With a bit of hacking around it should be able to handle any of the files.

The program processes the file in two parts, the headings and the details. This is done because the headings are where all the control characters are found, meaning they have to be processed a byte at a time, REAL slow. Version 1 of the program tried to do the whole file this way, and printed out status information every step of the way. I got my washing done and had lunch while it did the first 700

lines of the file, obviously too slow to be useful. Using a different method for the details and ditching the running status report means the whole job takes less than 6 minutes. Who says AmigaBASIC can't be useful?

To use the program, change the file names to point to your working copy of allchanges.hbin and the output files you want. The output files will need about 110k of disk space so they won't fit on the Extras disk. Besides, it's a good idea to try any new program on an expendable disk first anyway. Once you have done all this just run the program, six minutes later you will have the headings, about 8k, in the first output file and the detail text, about 100k, in the second output file. These files are pure text - you can type, print or ed them to your heart's content.

Feel free to use or modify the program in any way you like.

-- Bernd Kuenne

' Program to convert allchanges.hbin to text format

' Bernd Kuenne 14/2/87

```
start:
filein$ = "df1:hbin/allchanges.hbin"
fileout$ = "df1:text/changeheadings"
fileout2$ = "df1:text/allchanges"
lines = 0
delctr = 0
CLS
PRINT
PRINT "Name of input file   - ";filein$
PRINT
PRINT "Name of headings file - ";fileout$
PRINT
PRINT "Starting headings at - ";TIME$
PRINT
openfiles:
OPEN filein$ FOR INPUT AS #1 LEN=4096
OPEN fileout$ FOR OUTPUT AS #2 LEN=4096
OPEN fileout2$ FOR OUTPUT AS #3 LEN=4096
GOSUB firsttime
'
' the headings have lots of control chars so do them
' a byte at a time
WHILE lines < 191
char$ = INPUT$(1,#1)
IF ASC(char$) = 13 OR ASC(char$) = 10 OR ASC(char$) = 15 THEN
GOSUB writefile
ELSEIF ASC(char$) < 32 OR ASC(char$) > 128 THEN
delctr = delctr + 1
text$ = text$ + " "
char$ = " "
ELSE
text$ = text$ + char$
IF LEN(text$) = 4 AND text$ = " 1" THEN
text$ = ""
END IF
IF LEN(text$) > 10 AND RIGHT$(text$,10) = " 2" THEN
text$ = ""
END IF
END IF
IF LEN(text$) > 75 AND char$ = " " THEN
GOSUB writefile
END IF
```

```
WEND
PRINT "Control chars deleted - ";delctr
PRINT
PRINT "Lines output   - ";lines
PRINT
PRINT "Headings done at - ";TIME$
'
' Now for the real guts of the file
'
lines = 0
delctr = 0
PRINT
PRINT "Name of details file - ";fileout2$
PRINT
' skip some junk
WHILE NOT EOF(1) AND LEFT$(text$,7) <> ".Etopaz"
INPUT #1,text$
WEND
PRINT "Start of detail at   - ";TIME$
PRINT
WHILE NOT EOF(1)
' the rest of the file has the real details
LINE INPUT #1,text$
IF LEFT$(text$,7) = ".Etopaz" THEN
delctr = delctr + 1
ELSEIF LEFT$(text$,3) = ".##" THEN
delctr = delctr + 1
ELSE
PRINT #3,text$
lines = lines + 1
END IF
WEND
CLOSE
PRINT "Lines deleted   - ";delctr
PRINT
PRINT "Lines output   - ";lines
PRINT
PRINT "Run ended at   - ";TIME$
END
writefile:
'
' have to suppress some random double spacing
blank$ = SPACE$(LEN(text$))
IF text$ = blank$ THEN
blanks = blanks + 1
ELSE
blanks = 0
END IF
IF blanks < 2 THEN
PRINT #2,text$
lines = lines + 1
END IF
text$ = ""
RETURN
firsttime:
WHILE char$ <> "A"
char$ = INPUT$(1,#1)
WEND
text$ = char$
RETURN
```

Yet Another StripC Program by Peter Story

Why would anyone bother writing a program to strip comments from C include files? After all there is a public domain program StripC on AUG Disk #6. Well, as usual it became a matter of necessity. Software that I had written worked fine under Version 1.1 of AmigaDos, but always came up with a Guru message under Version 1.2. After getting the version 1.2 C include files, I was shocked to see compiler error messages from every C include file that I "included" in my program. I had used the stripped include files, as with a single drive system there's no way I can fit the compiler and the unstripped include files on the one disk. Well you guessed it, I didn't have a copy of Fred Fish disk #6 and the stripped include files had REALLY been stripped. In fact every include file I looked at was missing C source code from the front of the file and this was causing the compiler errors.

Talking About Comments

A C comment, for those of you who haven't yet looked at some C code, is anything beginning with /* and ending with */. The comments contain information useful to the programmer and hopefully anyone else reading his program. They are totally ignored by the compiler and only increase the time it takes the compiler to parse the file, as well taking up more disk space and increasing the time it takes to load from disk. The removal of C comments is normally only done on include files which contain a common set of declarations and constants. Using a common set of files with each separately compiled source module in a large program ensures that all source modules will be supplied with the same definitions and variable declarations. This eliminates the problem of using different definitions in some modules for the same object (which won't be detected by the compiler) and then, after linking the compiled modules, having the program crash. Finding this type of bug can be very difficult.

Include Files

Why are these common files called include files? Well, it's because to use them with a C source file you use the following line in your program.

```
#include "filename"
```

This causes the C compiler to load and scan "filename" instead of you having to put it into your source file when you write the program.

Back To My Tale

A closer look at the unstripped and stripped include files showed that any that had a comment box at the beginning like this:

```
/******
*
*           C comments live in here
*
*****/
```

had the comment box and all the C source up until the end of the next comment removed from the file.

What to do? I could have rung around to find who had a copy of StripC, but I didn't. Why not write my own stripping program - it should be pretty easy to search a file for comments and remove them. Several hours later, I had my own

stripping program. But since I had to reinvent the wheel, I decided to strip the file as much as possible. Consecutive spaces and line feeds are replaced with a single space or line feed and where a C comment appears on a separate line the complete line is removed. On the stripped include files I had, only the /* ... */ text was removed. This meant that comments like:

```
/* comment 1 */
/* comment 2 */
/* comment 3 */
```

left 3 unnecessary line feeds in the file. All this may seem a bit petty, but running my stripping program on the already stripped version 1.1 include files released an extra 25k bytes of disk space. Intuition.h (48k unstripped, 14k stripped) was reduced to 11k bytes after I stripped it. And the version 1.2 include files? No problems. I was now able to recompile my program with the 1.2 include files and it ran perfectly. The problem that caused the 1.1 compiled version to fail? I'm not really sure and I've been too busy to take the time to find out yet.

I've provided the source (strip.c) and executable (strip) for this program to AUG as public domain software for anyone interested. The program was written for the Lattice C compiler and uses its file handling routines. Not very efficient, but I was in a hurry at the time. I have some other public domain programs I've written which use only Amiga calls and this allows executable code of about 3k bytes instead of 15k bytes. That's using Lattice, tool. By the time you read this article all the programs will be available from AUG. The programs are all in C (sorry to all the Basic fans) and the articles will be of a technical nature. Now that I've committed myself I'd better give a brief list of the programs:

- C source code generator for the DiskObject and Image structures required for programs that generate icons for each file they create. (They should all do this but many don't.) Just design your icon with IconEd and this program will produce the C source for the icon structures.
- Demonstration program showing how to generate icons (.info files) for each file the program saves. The icons (.info files) are configured to reference the 3rd demo program and can be double clicked to automatically load this program.
- Demonstration program showing how to use the Workbench startup message to determine what icons were selected when the program was started and how to load those files if they belong to this program.

As you can see all three programs are related to the one subject - How to create programs that correctly use the Workbench interface. This will be the topic of my next article.

-- Peter Story

AmigaBASIC Serial Port Fix

After nearly going crazy trying to decipher the Rom Kernel manual on the serial device, I finally struck gold and came up with what I wanted.

Here is a Fix for allowing all ASCII Characters from 0 to 254 to be passed to AmigaBASIC when using OPEN "COM1:" etc. Previously AmigaBASIC would LOCK UP if a user typed a CTRL S, which is very bad for things like BBS software.

/* FixSerial.c Version 2.0

This program will set the serial device so that when AmigaBASIC OPEN's the COM port it passes through ALL the ascii characters including CTRL-S AND CTRL Q (Xon/Xoff). So Bulletin Board or Terminal Programs wont lock up.

Other Terminal Packages will reset the affects of this program ie) StarTerm, Online!, So if necessary use it again. AmigaBASIC does not touch the Ctrl Character options so the above is not necessary.

Compiled with MANX C using 32 bit option (+L) and Linking to 32 bit Libraries. (C32.LIB).

(C) 1987 By John Herks. May be used for non-commercial purposes Only!!!

*/

```
#define ESC 0x1B
#define COPYRIGHT 0xA9
```

```
#include <exec/types.h>
#include <exec/nodes.h>
#include <exec/lists.h>
#include <exec/memory.h>
#include <exec/ports.h>
#include <exec/libraries.h>
#include <exec/devices.h>
#include <exec/tasks.h>
#include <exec/io.h>
#include <devices/serial.h>
#include <stdio.h>
```

```
extern struct MsgPort *CreatePort();
```

```
int Set_Serial()
{
    int index = 1;
    ULONG value = 4294967295;
```

```
    struct IOExtSer IORser;
    int error;
```

```
    IORser.io_SerFlags |= SERF_SHARED;
    if ((error = OpenDevice (SERIALNAME, 0, &IORser, 0)) != 0) {
        printf("Can't Open Serial Device, Error=%ld\n",error);
        exit(1);
    }
```

```
    IORser.IOSer.io_Command = SDCMD_QUERY;
    if ((error = DoIO( &IORser ) != 0) {
        printf ( "Query Status Error %ld\n", error);
        exit(1);
    }
```

```
    if ((IORser.IOSer.io_Message.mn_ReplyPort =
        CreatePort( "SetSerial", 0 )) == NULL) {
        printf( "Unable To Create Port For IO Message\n" );
        CloseDevice( &IORser );
        exit(1);
    }
```

```
    switch( index ) {
        case 1: IORser.io_CtlChar = value; break;
        default: exit(1);
    }
```

```
    IORser.IOSer.io_Command = SDCMD_SETPARAMS;
```

```
    error = DoIO( &IORser );
```

```
    DeletePort( IORser.IOSer.io_Message.mn_ReplyPort );
```

```
    CloseDevice( &IORser );
```

```
    if (error) {
        printf( "Error %ld Doing IO To Set Params\n", error );
        exit(1);
    }
```

```
    if (IORser.IOSer.io_Error) {
        printf("Error %ld From Serial Device in Set Params\n",
            IORser.IOSer.io_Error);
        exit(1);
    }
    return 1;
}
```

```
main()
```

```
{
    int error;

    error = Set_Serial();

    if (error != 0)
    {
        putchar(ESC);
        printf("[7m");
        printf("FixSerial ");
        putchar(COPYRIGHT);
        printf(" 1987 By John Herks\n");
        putchar(ESC);
        printf("[0m");
        printf("May Not Be Used For Commercial Purposes\n");
        printf("Serial Port Patched For AmigaBASIC\n");
    }
}
```

```
exit(1);
}
```

Well, that's about it. On another subject, Commodore Australia just sent me some documents for the Developers Version 1.2 software for which they are asking \$30. I have promptly mailed my cheque but whether I will get anything is another matter.

-- John Herks

Unprotecting KickStart RAM

BIG NOTE: This modification was carried out on a NTSC Amiga, which has the daughter board installed, so this won't work on the PAL Amigas with the single board.

Here is some information on making your Kickstart RAM unprotected, which I have found to be Very Useful.

With this modification, when you're not interested in writing to the Kickstart RAM, you can switch the capability off and its write-protected just like a normal Amiga! Of course, this leaves the RAM open to crazy programs, but that simply means that if a program goes wild and happens to write to a critical area you have to turn off the machine to reboot Kickstart, rather than just inserting Workbench. To me, the ability to debug is worth it. Besides I can easily return to write protection.

First, a few words about how the Kickstart sequence works. When you turn on your Amiga the bootup ROM reads the Kickstart disk into the Kickstart RAM at \$fc0000. Then the 68000 does a read to the location \$f80000. (Actually, anything with address bits 19-23 high and A18 low would do). This condition is detected by a PAL called which DAUGCAS write-protects the Kickstart RAM.

Before we continue: A small disclaimer

1. I can't be responsible for anything you do to your Amiga. I believe I've explained this thoroughly enough so someone with soldering experience and a reasonable mechanical ability can do it without much difficulty, but you are taking your computer into your own hands.
2. Commodore may have or may in the future change the layout of the circuit board or change the PAL's, making these instructions obsolete. If your board doesn't look like what I'm describing, DON'T DO ANYTHING.
3. This will void your warranty.
4. I represent no company or organization in any way whatsoever.

The Modification

Simply install a switch to force the write protect signal (WPRO*) inactive. This is the method I used, and is really simple because C-A graciously left two holes in the circuit board for the wires to my switch. It could probably be removed (in case you had to take it in for servicing at a later time) without leaving any traces. Well, actually you would have to drill a hole in the back panel to mount the switch.

UNPLUG YOUR AMIGA!!!! It will be very easy to strike a metal object (soldering iron, solder, resistor, wire, watch, etc) across two consecutive traces and who knows what might happen.

First, you need to take off the cover of your Amiga and take off the metal shield that covers the circuit boards (yes, all 19 screws and 2 twist tabs). Don't forget the little push tabs on the sides of the Amiga, so you can take the top cover off. Admire the graffiti on the cover while you're at it.

So that we agree on orientation, the front of the Amiga is towards you, the power supply is on the left, and the parallel port is on the upper right side. I will refer to the rear of the Amiga as the "top."

Inside, you'll find a main mother board, and an L-shaped "daughter board" which is upside down, above the main board. The PAL is located on the daughter board at location J-6. Its on one of the inside corners, next to the letter "J" and has the label "DAUGCAS" alongside it. This is a 20 pin chip.

There should be a white outline of the chip between the pins with a notch at one end. From the notch, count pins clockwise until you get to pin #13, which is the third one from the upper right corner of the chip. Pin 13 is the WPRO* signal.

Originally, the WPRO* signal went through a resistor and LED to +5V. Since WPRO* is active low, the LED would go on during bootup while the RAM was being written, and then go off at about the time it told you to insert the Workbench, and remain off until you turned the machine off.

Well, it's silly to have an LED inside a closed box, so C-A decided to save costs and not insert the resistor and LED, but the circuit board should still be the same. I don't know on how many boards the LED was inserted, or on how many boards the LED socket will still be fitted, so inserting the wire to the switch may be a slightly different procedure for some of you.

If you follow the trace from pin 13 of DAUGCAS, it will go to one of the resistor terminals. Insert the stripped end of a wire into this hole at the left end of the resistor. (Just heat the hole with your soldering iron and push the wire through).

As a precaution, before working on the board, put a piece of paper under where you'll be soldering so solder splats don't hit the board below. I didn't get anything on my paper, but it's good protection.

Above the resistor space is a white box marked "1" with two solder holes. Insert the stripped end of another wire in the left-most of these two holes inside the box. You now have the two wires necessary to run to your switch: one is connected to WPRO* and the other to +5V. With the switch "on" you will pull WPRO* hi (inactive) and make the RAM writable.

A note about mounting of the switch. I suppose you will want to mount this in the back panel. This panel is removable, so it's easy to put it flat on a table and drill a new hole. Before you remove the back panel and drill the hole, however, put the metal shield back on (screws not necessary) and hold your switch in the location you wanted. If it's like most switches, it probably won't fit next to the metal shield UNLESS it's as far right as possible (over the keyboard jack). I suggest you use a high quality miniature switch. Now, go drill the hole.

When cutting the wires, leave a few inches of slack so they can fit through the crack in the metal shield in the upper right corner. (Convenient, huh?)

Now put it back together and you have switchably-writable Kickstart RAM!

A Note About Operation

For normal operation, leave the switch "off" when you power up. This will look and act exactly like a normal Amiga. To enable writing the RAM, turn the switch "on". If, after you have been writing the RAM, you decide to write protect it, the switch alone won't do it - TURNING THE SWITCH BACK OFF STILL LEAVES THE RAM WRITABLE. You need to either do a read to location \$f80000 or turn the machine off and on again.

If you want the facility to turn off WPRO* with a switch, you can use a 3 pole switch with WPRO* in the middle terminal and +5V on one side and GND on the other, but there isn't a convenient hole for GND so you'd have to find one on the board. This would have an immediate reaction with any direction of the switch.

Technicalities

All of the above works, but if you want to know a little about how, read on. The WPRO* signal is actually both an output and input of the DAUGCAS PAL. The logic used to generate it has what we call a "set" term and a "reset" term. It acts like a S/R flip-flop. The "set" term makes WPRO* active when the processor does a read cycle from an address whose bits 19-23 are high and bit 18 is low. The "reset" term comes from the hardware reset line that is active when the machine powers on.

Once WPRO* is set, it won't be reset until power on again, and once reset it won't be set until that read instruction. WPRO* is both an output and an input of the PAL. The output is generated from the above equation, and the input is used to disable a write signal from ever getting to the RAM.

The very astute of you might criticize the forcing high of a low output. Well, the only time the output is forced is during the read cycle (a couple hundred nanoseconds), because after that WPRO* is tricked into thinking it was reset and the output agrees with what is being forced so no contention. The PAL used, according to the manufacturer, is able to handle voltage applied to outputs up to Vcc, which is +5V, so we're within specs.

Another modification that I have in my machine is a piggy backed chip which will allow to make DF3: (5.25" disk drive my boot drive). I will probably describe this modification at a later date.

-- John Herks

Interesting Stuff

Welcome to our biggest issue yet! Four more pages than normal, and we could grow even larger if you keep on writing articles. On behalf of all our readers, I'd like to thank everyone who took the time to put pen to paper (or fingers to the keyboard!) to write articles. Thanks.

The big news this month seems to be the "release" of AmigaDOS version 1.2. I use the term "release" loosely, because you can't actually go into a store and buy the **Amiga Enhancer** package, because Commodore Australia haven't got any, even though it has been available since December in the USA. What you **can** buy is the developer's seven-disk set, for anywhere between about \$35 and \$65 depending on which dealer you get it from. I also have it on good authority that several of the larger dealers will **give** you a copy of the Kickstart, Workbench and Extras disks if you bought your machine from them.

The other big news is the **Amiga 500** and the **Amiga 2000** machines. These machines **do** exist, Commodore Australia has at least one of each machine, and the serial number of their Amiga 2000 seems to suggest that there are at least 200 of them somewhere. Commodore will be showing the machines to developers, dealers and interested parties at the Sydney PC show at the end of March, by invitation only. All developers (and probably dealers) received specifications on the new machines a few days ago in the mail. **Both** new machines still have 68000 processors, the A500 comes with 512k and the A2000 with 1Mbyte standard memory.

The Amiga 500 is pretty close to what we have now, except that it can have up to 1Mbyte of ram and a battery-backed clock/calendar internally, and the keyboard is integral to the main unit (ie it's all one piece, like a Commodore 128). Kickstart is in ROM, and the power supply is external, probably to make it easier to get approval in the various countries where it will be sold. Both new machines use the current co-processor chips, meaning that they can still address only 512k.

The Amiga 2000 is has 5 internal zorro slots, 2 IBM/PC and AT slots and 2 IBM/PC slots. These four slots are driven by the Amiga PC BridgeBoard, which **does not** come fitted as standard. Unique to the A2000 is a "video slot" for an internal NTSC or PAL encoder for composite video or internal genlock. The machine has fittings for two 3.5" and one 5.25" drive. The A2000 also has Kickstart in ROM and comes with a battery-backed clock/calendar

Commodore also mentioned that they have negotiated a deal with Lattice to allow developers to upgrade their 3.03 C compilers to version 3.10, at an intended price of \$50 to \$100. The new Lattice is at least as good as Aztec C, and comes with a new 450 page manual. If you ever tried to find anything in the old manual, you'll find the new one a pleasant change. The compiler itself is 41k smaller; it compiled itself to that size despite many new features. Commodore will notify developers when the upgrade is available.

Commodore also announced that the Amiga Developers Conference, due to be held in a month or so has been postponed, because the relevant Commodore staff would be too busy. The letter also attempted to explain why it takes Commodore so long to send out orders after they have the money. Explained, yes. Justified, **NO!** Really, Commodore, you'll have to do better than 3 weeks for a pre-paid order. Actually, even being able to get things in 3 weeks is better than the time it usually takes!

If everything goes just right, we'll be showing an instructional video-tape on AmigaDOS, the Workbench and the CLI at the March meeting. The tape still hasn't been converted as I write (22-Feb-87), so I haven't seen the tape myself just yet. We'll get the tape in the next few days, but the newsletter has to be at the printers in the morning, so I can't write a review. If you are new to the Amiga, come along to the March meeting and see the tape.

Our Software Librarians have asked me to point out that disk orders accompanied by blank disks usually get filled faster than those where we supply the disks. At the moment, our disk supplier is out of stock, so there will be a small delay for those orders waiting on disks. If the demand is there, we might be able to supply public domain disks on 80 track 5.25" Amiga format. Since we don't sell 5.25" disks, you'll have to supply your own. Let us know if this service would be of any use.

I put my foot in it again last month over the PC-DOS to AmigaDOS utilities in AmigaDOS V1.2. I was wrong, the developer's disk set **does** have utilities for reading, writing and formatting IBM PC-DOS disks. The public release **Enhancer** does not have these utilities, though. Almost every time I make a statement or quote a price, it turns out to be wrong. Still, that probably won't stop me from doing it again!

I look forward to seeing all of you at the March meeting.

-- Peter Jetson

Member's Messages

Member **Peter Evans** is interested to hear of any genealogy/family history programs available on the Amiga. Please phone 584-2765 if you can help.

Roland Seidel is interested in forming a **Music Sub-Group**, and can be contacted by phone on 890-3934. Roland would like to have a small meeting of interested people at 1pm, Sunday March 8th, (yes, 1 hour before the AUG meeting), at his house, **32 Monash Street, Box Hill, 3128**. Might be worth a ring in you are thinking of coming, in case hundreds of people turn up!

Jim Eadie has offered to help people with C, etc, and can be reached on (059) 64 6771. That's at Launching Place, not far on the other side of the Dandenongs.

(I'd like to apologise to anyone who has asked to be included in this section, but hasn't made it to print yet. I seem to have lost a list that people wrote on at a recent meeting. If you want a message to appear here, put it in writing and send it to the PO box, or give it to me at a meeting.)

Another Startup-Sequence

Here is a tip on a good startup sequence for your Amiga, incidentally it is the startup sequence I use. The advantage to this startup sequence is EASY access to both the workbench and command line interface simultaneously.

I only recommend this tip for people with a 512K Amiga because it uses about 3K of memory, and people with 256K Amigas need every last byte.

This is the startup-sequence file:

```
newcli "con:540/150/100/50/CLI "
LoadWb
endcli > nil;
```

SOFTWARE ORDER FORM

Disk numbers :

Disks supplied by Amiga User Group @ \$10								\$	
Disks supplied by member @ \$2								\$	
Club Use Only									
Receipt #:				Mailed on: / /				Total \$	
Mail to: Amiga Users Group, PO Box 48, Boronia, 3155, Victoria.									
Member's Name:									
Address:									

Application for membership of The Amiga Users Group Inc

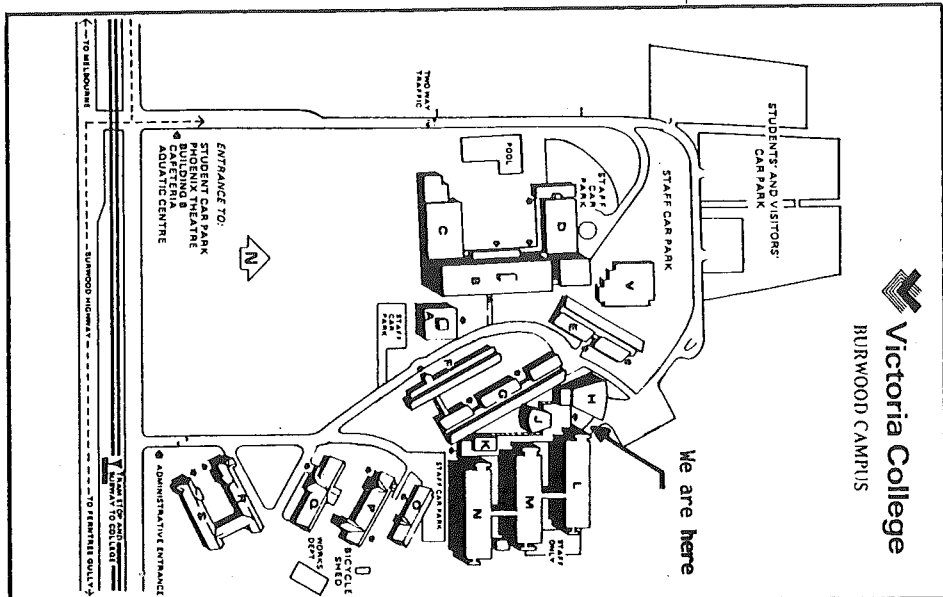
Membership is \$20 per year. Send your cheque to: **Amiga Users Group Inc, PO Box 48, Boronia, 3155**

Surname: _____ Details on this side are optional
 First name: _____ (no initials) Year of birth: _____ Do you own an Amiga: _____
 Address: _____ Occupation: _____
 _____ Postcode: _____ Interests: _____
 Phone Number: _____ STD Code: _____
 What services would you like AUG to provide: _____
 _____ Dealer's Name: _____
 _____ Dealer's Address: _____
 _____ Postcode: _____

Signed: _____ Date: _____
 In the event of my admission as a member, I agree to abide by the rules of the Association for the time being in force.

Are you happy with your dealer: _____

Club Use Only	Date	Paid	Rcpt #	Memb #	Card Sent
---------------	------	------	--------	--------	-----------



Where is Victoria College Burwood Campus?


New members and visitors sometimes have trouble locating our meeting place the first time. Victoria College is on the North side of Burwood Highway, Burwood, just East of Elgar Road. Coming from the City, turn left at the first set of traffic lights after Elgar Road. Follow the road around past the football oval, over three or four traffic bumps to the car parking areas near the netball courts. Further up the road, to the left, you'll find Lecture Theatre 2. That's us!

If you have a Melways, try Map 61 85.

March 1987 Amiga Workbench

P.O. Box 48, Boronia, 3155, Victoria, Australia

AMIGA™ Users Group

 **AMIGA Workbench**
Registered by Australia Post
Publication No: VBG7930
If Undeliverable, return to:
PO Box 48, Boronia, 3155
Victoria, Australia

Postage Paid
Caulfield
East 3145

+ Member # 10297 Expires end of 11-87

Campbell Gray
RMIT Comm Eng Dept
Box 2476V
Melbourne 3001