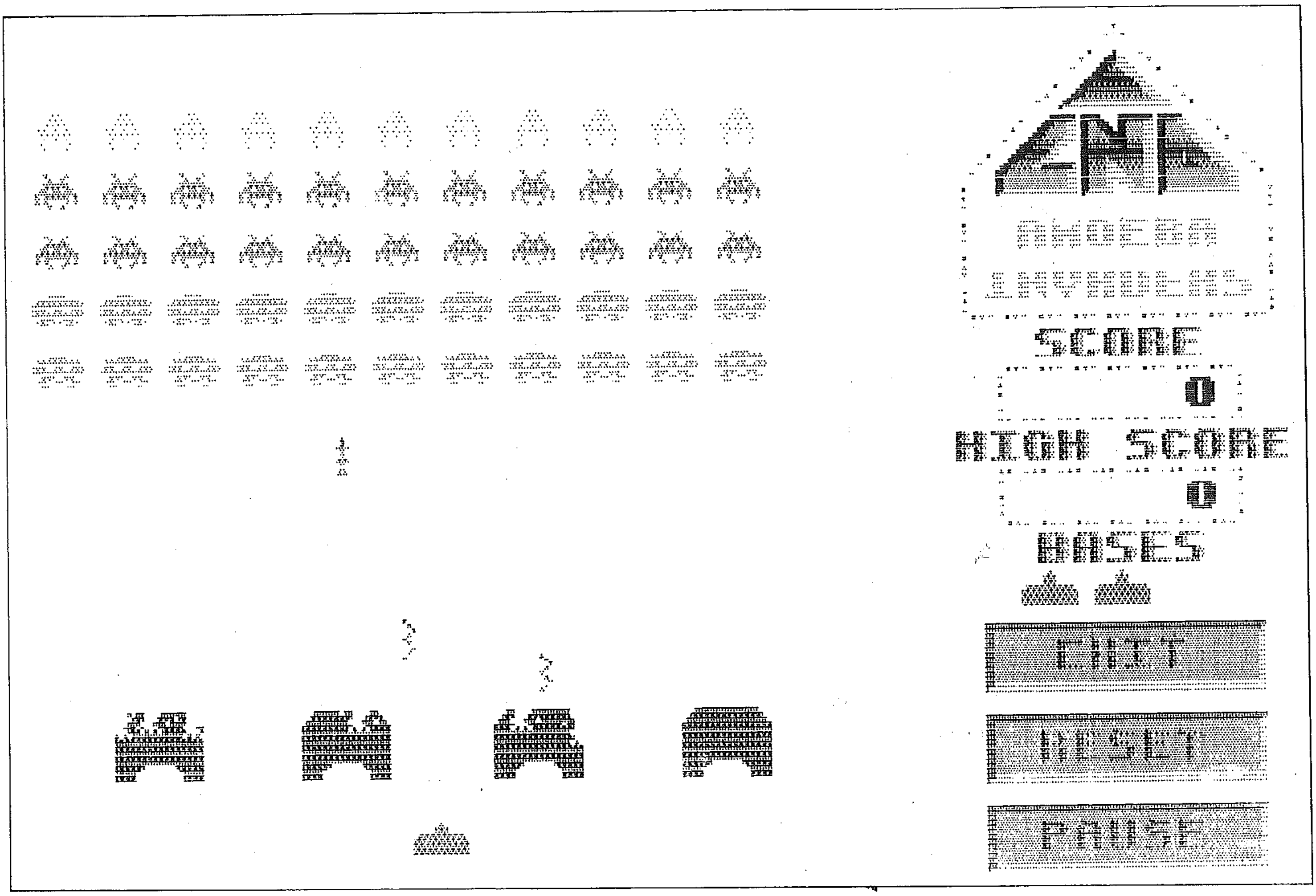
AMIGA

WORKBENCH

$2
RRP

Registered by Australia Post — Publication No. VBG7930

Number 21 Circulation: 1000 February 1988



Next AUG Meeting

Saturday, February 13th, 1988 at 2pm

AUG meetings are held in the Rotunda at Monash University
Wellington Road, Clayton Melways map 70 reference F10 and map 84A

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

Australia's Largest Independent Association of Amiga Owners
The *Amiga Users Group Inc* has no affiliation with Commodore

Who Are We?

The Amiga Users Group is a non-profit association of people interested in the Amiga computer and related topics. With almost 900 members, we are the largest independent association of Amiga users in Australia.

Club Meetings

Club meetings are held at 2pm on the second Saturday of each month in the Rotunda at Monash University, Wellington Road, Clayton. Details on how to get there are on the back cover of this newsletter. The dates of upcoming meetings are:

Saturday, February 13th at 2pm
Saturday, March 12th at 2pm
Saturday, April 9th at 2pm

Production Credits

This month's newsletter was edited by Peter Jetson. Equipment and software used was: TurboDOS S-100 computer, Brother HR-40 printer, Gemini 10x printer, Wordstar, Fancy Font and Grabbit.

Copyright and Reprint Privileges

Contributions

Articles, papers, letters, drawings and cartoons are actively sought for publication in Amiga Workbench. Please submit your contributions on disk, since that means they don't have to be re-typed! All disks will be returned! Please save your article in **text-only** format (If it can be loaded by ED, it is text-only). Absolute deadline for articles is 16 days before the meeting date. Contributions can be sent to: The Editor, AUG, PO Box 48, Boronia, 3155.

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

Public Domain Software

Disks from our public domain library are available on quality 3.5" disks for $8 each including postage on AUG supplied disks, or $2 each on your own disks. The group currently holds over 160 volumes, mostly sourced from the USA, with more on the way each month. Details of latest releases are printed in this newsletter, and a catalog disk is available.

Member's Discounts

The **Amiga Users Group** negotiates discounts for its members on hardware, software and books.

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!

Back Issues of Newsletter

All back issues of Amiga Workbench are now available, for $2 each including postage. Back Issues are also available at meetings.

AmigaLink - Our Bulletin Board System

The Amiga Users Group operates a bulletin board system devoted to the Amiga, using the Opus message and conferencing system. AmigaLink is available 24 hours a day on (03) 792 3918, and can be accessed at V21 (300bps), V22 (1200bps) or V23 (1200/75bps), using 8 data bits, 1 stop bit and no parity.

AmigaLink is part of the world-wide Fido/Opus network of bulletin boards, and we participate in the national and international Amiga conferences. AmigaLink has selected Public Domain software available for downloading, and encourages the uploading of useful public domain programs from its users. AmigaLink is FidoNet node number 631/324.

Newsletter Advertising

The Amiga Users Group accepts commercial advertising in Amiga Workbench subject to the availability of space at these rates:

Quarter page	$20
Half page	$40
Full page	$70
Double page spread	$120

These rates are for full-size camera-ready copy **only.** We have no photographic or typesetting facilities. Absolute deadline for copy is 16 days before the meeting date. Send the copy and your cheque to: The Editor, AUG, PO Box 48, Boronia, 3155, Victoria.

Amiga Users Group Committee

Co-ordinator:	Bob Scarfe	376 4143	Kensington
Vice Co-ord:	Fergus Bailey	211 7845	Malvern
Meeting Chair:	Ron Wail	878 8428	Blackburn
Secretary:	John Elston	375 4142	M' Ponds
Treasurer:	(temporarily vacant)		
Membership:	Neil Murray	794 5683	Dandenong
Purchasing:	Bohdan Ferens	792 1138	Dandenong
Book Library:	Joan Wood	580 7463	Aspendale
Disk Library:	Geoff Sheil	578 8362	Brighton
	Margaret Bedson	578 8362	Brighton
Editor:	Peter Jetson	762 1386	Boronia
SMAUG Co-ord:	Roland Seidel	890 3934	Box Hill

The Amiga Virus

by Bill Koester
Commodore-Amiga Technical Support

When I first got a copy of the Amiga virus, I was interested to see how such a program worked. I disassembled the code to a disk file and hand commented it. This article will try to pass on some of the things I have learned through my efforts.

Definition

The Amiga virus is simply a modification of the boot block of an existing DOS boot disk. Any disk that can be used to boot the Amiga (ie Workbench) has a reserved area called the boot block. On an Amiga floppy, the boot block consists of the first two sectors on the disk. Each sector is 512 bytes long, so the boot block contains 1024 bytes. When Kickstart is bringing up the system, the disk in drive 0 is checked to see if it is a valid DOS boot disk. If it is, the first two sectors on the disk are loaded into memory and executed. The boot block normally contains a small bit of code that loads and initialises the DOS. If not for this boot code, you would never see the initial CLI. The normal boot code is very small and does nothing but call DOS initialisation. Therefore, on a normal DOS boot disk, there is plenty of room left unused in the boot block.

The virus is a replacement for the normal DOS boot code. In addition to performing the normal DOS startup, the virus contains code for displaying the virus message and infecting other disks. Once the machine is booted from an infected disk, the virus remains in memory, even after a warm start. Once the virus is memory resident, the warm start routine is affected - instead of going through the normal startup, the virus checks the boot disk in drive 0 for itself. If the virus in memory sees that the boot block is not infected, it copies itself into the boot block, overwriting any code that was there before. It is in this manner that the virus propagates from one disk to another. After a certain number of disks have been infected, the virus will print a message telling you that Something Wonderful has happened.

Dangers

When the virus infects a disk, the existing boot block is overwritten. Since some commercial software packages and especially games store special information in the boot block, the virus could damage these disks. When the boot block is written with the virus, any special information is lost forever. If it was your only copy of the game, then you are out of luck and probably quite angry!!

Mechanics

Here is a more detailed description of what the virus does. This is intended to be used for learning and understanding ONLY! It is not the author's intention that this description be used to create any new strains of the virus. What may once have been an innocent hack has turned into a destructive pain in the #$@ for many people. Let's not make it any worse!

Infiltration

This is the first stage of viral infection. The machine is brought up normally by reading the boot block into memory. When control is transferred to the boot block code, the virus code immediately copies the entire boot block to $7EC00, and then JSR's to the copied code to wedge into the CoolCapture vector. Once wedged in, control returns to the loaded boot block which performs the normal DOS initialisation. Control is then returned to the system.

Hiding Out

At this point, the CoolCapture vector has been replaced, and points to code within the virus. When control is routed through the CoolCapture vector, the virus first checks the left mouse button, and if it is down, the virus clears the CoolCapture vector and returns to the system. If the left mouse button is not pressed, the virus replaces the DoIO code with its own version of DoIO and returns to the system.

Spreading

The code so far has been concerned only with making sure that at any given time, the DoIO vector points to the virus code. This is where the real action takes place. On every call to DoIO, the virus checks the io_Length field of the IOB. If this length is equal to 1024 bytes, then it could possibly be a request to read the boot block. If the io_Data field and A4 point to the same address, then we know we are in the strap code and this is a boot block read request. If this is not a boot block read, the normal DoIO vector is executed as if the virus was not installed. If we are reading the boot block, we JSR to the old DoIO code to read the boot block and then control returns to us. After reading, the checksum for the virus boot block is compared to the checksum for the block just read in. If they are equal, this disk is already infected, so just return. If they are not equal, a counter is incremented and the copy of the virus at $7EC00 is written to the boot block on the disk. If the counter ANDed with $F is equal to 0, then a rastport and bitmap are constructed and the message is displayed.

Ha Ha

< Something wonderful has happened >
< Your AMIGA is alive!!! >
< and even better >
< Some of your disks are infected by a VIRUS >
< Another masterpiece of the Mega-Mighty SCA >

Prevention

How do you protect yourself from the virus?

1) Never warm start the machine, always power down first. (Works, but not too practical!)
2) Always hold down the left mouse button when rebooting. (Also works, but only because the virus code checks for this special case. Future viruses may not!)
3) Obtain a copy of VCheck1.1 and check all disks before use. If any new viruses appear, this program will be updated and released into the public domain.

Just like a real virus, the best course of action is education and prevention!

[Editor's note: This information has been directly taken from a printout of a UseNet message provided to me by an AUG member, Stephen Campbell. Version 1.2 of the VCheck program is available on Fred Fish disk #126, and the software library people have several other similar anti-virus programs.]

BaSIG, Our BASIC Special Interest Group

by John Elston

Last month, I regaled you with my attempts to set up the BASIC special interest group. Since then, through a series of subtle threats or blackmail, I have convinced several of the people who were foolish enough to attend the last two meetings to act as a "committee". This committee, primarily, will organise the SIG so that future BaSIG meetings will be planned, coordinated and generally have some form of controlled direction.

As the originator of the BaSIG, I have my own concept of what a BASIC SIG should provide; however I expect that other people will have different views and I hope that the committee will represent a cross-section of various levels of expertise and diverse views concerning BASIC and its use.

As I see it, BaSIG exists to promote the use of Amiga BASIC, through discussion, example and use. One of the things I would like make available to members of BaSIG is a disk of public domain BASIC programmes generated by BaSIG and a disk of collated programmes from the Fred Fish, Amicus and the Amigan public domain disks. The latter I am trying to organise at the present time, and the former requires input from anyone who has written BASIC programmes and is prepared to put them forward to be used to compile a BaSIG public domain disk. If you have a programme you think may be of interest to others, my telephone number is below, please contact me so we can arrange for a transfer of your programme or catch me at the next BaSIG meeting.

At the next meeting, I hope to present a short introduction to AmigaDOS in relation to AmigaBASIC and show how to "trim the fat" off a standard disk to get more free space, how to use system level devices such as the parallel and serial ports, with particular reference to single drive machines. In the future, I hope that we can persuade people who have had experience of various aspects of BASIC to give us the benefit of their trials and tribulations.

As usual, if you have any comments, enquiries or suggestions about BaSIG, I will be only to happy to chat to you, my phone number is (03) 375 4142.

The Wedge Hard Disk System

by Peter Jetson

The Wedge is a small adaptor assembly that fits on the side of an Amiga 1000 or 500, and allows the use of an IBM hard disk controller and hard drive on the Amiga. The Wedge comes with driver software configured for 10, 20 and 40 megabyte drives.

Besides the Wedge itself, you'll also need a Western Digital WX1 or DTC5150 IBM hard disk controller, an ST506 hard disk drive, a power supply and cables. Unless you're happy with naked disk drives, boards and power supplies on your computer desk, you'll probably also want some kind of enclosure. This is not the hard disk system for you unless you are technically competent. The work you do to get your hard disk running is what saves you the money.

The Wedge costs $175 Canadian, add another $10 Cdn for Airmail postage. That comes out at about $200 Australian, and Customs charges could add another $70 or so to that. Cheap IBM hard disk drives and controllers are about $600 or so, and a power supply could cost you another $100. Forgetting about the case, that adds up to about $1000.

Technically, the Wedge is quite simple, consisting of an 86 way connector into the Amiga, five ICs that do the buffering and address decoding, and an adaptor board with a 62 way IBM PC bus connector into which you plug your IBM hard disk controller, and into that you plug your hard drive. The Wedge seems to work by hard-wiring the lines on the IBM hard disk controller to make it look like an IBM is selecting it, and decoding a small portion of the Amiga address space to enable the board. The read/write, low order address lines and data lines and buffered and connected though to the Amiga. Thus, the registers in the IBM controller end up as a few bytes of memory to the Amiga.

The Wedge is powered directly from the Amiga, but since a typical hard disk drive could draw up to 1 amp at 5 volts and a startup current of up to 2.5 amps at 12 volts, the hard disk itself must be externally powered. The Tandon 3.5 inch TM362 I chose for the job normally consumes 0.6A at 5V and 0.8A at 12V with a startup current of 2.0A at 12v for about 10 seconds. The 40 Watt switching power supply I am using (an old Astec brand from a Tandy model 3 computer!) will supply about 2 amps at 12 volt and 5 amps at 5 volts.

The (miniscule) documentation that comes on disk with the Wedge mentions that the Wedge has been tested with, and works well with the Western Digital WX1 controller and the DTC5150 controller. It works, but 2.5 times slower with the NCL controller. It may work with others, but who knows? Fortunately, a lot of the cheaper IBM hard disk and controllers on sale here come with the DTC controller.

After all that technical rhubarb, I'm sure you're interested in how well (or otherwise) it performs. In a nutshell, it's great! As the following table shows, you can expect your Amiga to read from the hard disk about 2.6 times faster than floppy, and write about 4.2 times faster. The increase in performance is not as pronounced as on some other computers, but still welcome. When Commodore provides a new hard disk file system (which they are widely rumoured to do), you can expect a further speed increase of about 5 times.

The following table contains performance timings using Rick Spanbauer's DiskPerf program from Fred Fish disk #48. The final column shows how much faster the Wedge is than the normal Amiga floppies.

The C Ltd hard disk drive is included for comparison because it is generally regarded as the highest performance Amiga hard disk system. The C Ltd system uses DMA, the Wedge doesn't. As you can see, the Wedge performs quite well against its peer.

The Wedge system tested uses a 3.5" 20 Mbyte Tandon TM 362 stepper-motor hard disk drive. The Amiga 1000 used for testing is stock-standard except for a KickStart in ROM modification, which provides an extra 256k of fast RAM.

Test	Amiga Floppy	Amiga C Ltd	Wedge/ Tandon	Times Faster
File Creations	0.1	6	8	80
File Deletions	1	10	18	18
Directory Scan	36	51	52	1.4
Seek + Read	17	55	54	3
Read 512 bytes	11702	28807	30481	2.6
Write 512 bytes	4818	12977	20641	4.2
Read 4096 bytes	12365	34952	32363	2.6
Write 4096 bytes	4974	17712	20971	4.2
Read 8192 bytes	12423	34952	32363	2.6
Write 8192 bytes	4955	17712	21487	4.3
Read 32768 bytes	12483	34952	32768	2.6
Write 32768 bytes	4964	18078	20805	4.2

The Amiga Users Group is negotiating for distribution rights with the Canadian company which markets the Wedge. We'll let you know more as details come to hand. If you can't wait that long, see me at the February meeting and I'll give you the details so you can get your own. However, waiting may well get you a better price.

My Wedge system will be on show at the February AUG meeting.

Review: The Proton Disk Drive

by David Peel

This 3.5" external disk drive for the Amiga can be summed up in one word - Excellent! It is quiet, compact and solidly reliable.

The internal drive in my Amiga 1000 sounds like a lion roaring with rage next to this little wonder. I say little, to be precise I mean that it is 10cm wide x 3.5cm high x 19cm long. This means that it can slip under where your keyboard fits or it can happily sit beside the monitor on top of the main unit. It is solidly built with a metal case almost the same colour as the case of the Amiga. The flat ribbon cable is firmly anchored at each end and the connecting plug is of good quality. There is a through socket outlet on the body of the unit at the rear to enable one to daisy chain drives.

I bought mine for $350.00 from DownUnder Software in Richmond, although I think that might have been an introductory offer.

Finally, let me say that I am delighted because it is an Australian product, made in South Australia. If their memory expansion is as good that's where I'll be headed. I suppose that I am making one of those basic discoveries, namely that there is existence of a sort with one disk drive, but definitely, life of a better sort with two.

Introduction to the Programming Language `C'

by Eric Salter

I gave an introductory talk on this topic at the November '86 MICOM meeting, but, as with quite a few things, I had no time to prepare myself and do justice to this exquisite and precise programming tool. Hence after some discussion with Ralphe Neill, we both decided that it would be a good thing to run a series of articles on C, building on each until we had covered the main features of this language. These articles were first published in the Journal of the Microcomputer Club of Melbourne MICOM. With the number of phone calls I get each week on the topic of 'C' programming for the Amiga, I thought it about time to revise them with special emphasis on 'C' in the AmigaDOS environment. We envisage running a series of workshops, discussions, question times or whatever on C during 1988 and this column will attempt to answer questions that were raised at those meetings in a simple and hopefully informative way so that you will feel encouraged to go out and learn C in the best possible way - by writing in it!

C is a general-purpose language. It is extremely powerful as one can express logical ideas in concise expressions which in turn make for effective use of computer system resources - in terms of program development time, speed and ease of programming, and in terms of computer execution and memory usage. It is not a high level language, more akin to assembler than to Pascal, but it incorporates many of the features of structured programming which are found in higher level languages such as Pascal. One line of C code, has more of a one to one relationship to the final machine code than does the equivalent in say Pascal or BASIC. Economy of expression is therefore encouraged - a small source file will be a small object file (usually) and vice versa. These features then, make C a nice language to do things that would be prohibitive, in terms of overhead, in other languages. Its power brings with it some cautioning words. It is a dangerous language, because of its freedom of expression, absence of restrictions, lack of type and stack checking, which is left up to the programmer's acumen. So powerful is it, that the UNIX operating system, which has about 13000 lines of code, was written almost entirely in C, with 800 or so lines of code in native assembler code (The latest revision has considerably more). Similarly, the Amiga's operating system, was for the most part, written in C except for highly optimized assembly code for some of the ROM Kernel routines and low-level device drivers. So, in terms of today's "Home" computers, it is a good idea to become fluent in this wonderful language.

C was written by Dennis Ritchie of Bell Labs's for implementation under Bell's UNIX operating system on the DEC PDP-11 computer. C was based on BCPL via a language called B, however as the original K & R book mentions, it was not a dialect of B or BCPL for that matter. C is a compiled language. In BASIC, each line of your BASIC program is interpreted so each time the interpreter sees a BASIC statement, it has to look at it and translate it into many thousands of actual machine operations, even if the statement has been seen before. In C the source file is passed through a program, called a compiler, which converts your C code into either Assembly language or directly into the native machine code of your computer. This means that your program will be many times faster than the equivalent in BASIC (that is if you could even code the task in BASIC). The disadvantage of a compiled language is that you have to wait 'till the compiler finishes the translating before you can run it, and any logical errors in it will usually crash the machine, rather than, in BASIC, tell you the line in which the error occurred. You cannot, therefore interrogate the C compiler, as you can in BASIC, and look at the values of the variables that were active at the time of the malfunction. The result is, that if you are a sloppy programmer, you may spend more time debugging your code and waiting for the blasted machine to re-compile before you can test it again. You can be a sloppy programmer in BASIC - but you can get away with it more easily.

First Steps

To successfully compile a simple working C program is the most difficult thing to do for a beginner. The cynics will say this remains true regardless of experience. Let's look at the process of getting a simple program running without regard to how to how to program in C.

```
#include "stdio.h"

main()

{
     printf("Nothing Happens\n");
}
```

Above, we have a simple C program. We will call it "xyzzy.c". The object of our program is simple:- when the user types `xyzzy' on the terminal, our program will run and print `Nothing Happens' on the screen.

Step 1 - Creating the program

A `C' program is nothing more than text. To be precise, it is an ASCII text file which may contain only printable characters of the ASCII codes including spaces and tabs. To create a `C' program you need a text editor of some sort, preferably a screen editor. On the Amiga, you already have a screen editor in the form of `ed'. It is the bare minimum for writing `C' source code and you would not use it for serious source code editing. There are some excellent full screen editors around for the Amiga including some of the public domain programs such as Emacs. Alternatively, you may go for a commercial version which may come with your `C' compiler e.g. Manx Aztec's `z' which it their equivalent of the UNIX screen editor `vi', or Lattice's screen editor sold separately at an exhorbitant price. Some of my early work with `C' was on 8-bit machines and there I used WordStar in non-document mode. Which ever method you use, make sure you are happy with it. Having created the above file and called it "xyzzy.c", the next step is to compile the source. With the latter versions of the Lattice C compiler, there is an easy way to compile a program with many of the options that the compiler provides, set to default values. Most of these default values are adequate until you are writing code that depends on various features of the compiler. With Lattice, the command to compile our program is:

```
lc xyzzy
```

This line asks the Amiga to load the compiler and then passes it the name of the program to compile, in this case, ours! The compiler then does a lot of thinking and thrashing of disks and several minutes later (yes it can take quite a long time), it informs us that our code has been compiled and it is so many bytes long. Oh, you mean it didn't compile for you, there were thousands of error messages and warnings. Hmm, not as easy as you might think eh what? The trick to successful compilation is setting up your programming environment so that you and the compiler know what each is expected to do. The commonest cause of compile-time errors is the compiler not being able to find a particular include file or the source file or something. We will look at the programming environment in detail later in the series. Now, if you run an Aztec C compiler, the code will be compiled into another ASCII text file but this time in Assembly language, requiring another step to assemble the code to machine code ready for linking, which we will examine next. The Lattice compiler on the other hand, will compile directly to machine code, leaving out the assembly language step. The advantage of having an assembly language step is that you can see how the C compiler translates your code into assembler. When you know what is happening, you might then alter the assembly code to do things more efficiently than the C compiler - this is called hand optimizing and is usually done for speed. If you really need speed, you should be programming in assembler first off, or writing the time-critical parts of your program in assembler and then linking them into you C code.

With the compilation complete, the resulting file generated by the compiler will still not run - because it is not a machine code file. It is a file which contains some code and information for a program called a LINKER, which will link into your program certain routines which are found in standard libraries or in libraries that you make yourself. These routines, such as those for opening a file, doing string compares, concatenation etc, are not part of the standard C language, but are found in virtually every implementation of C and add to its usefulness. The routines may be written in C itself and have been compiled already, or they may be in assembly language and be the product of a special assembler producing linker-compatible files which can be linked into your C program. To link a file usually involves knowing which libraries are needed and specifying them on the command line of the linker. The linker requires that the program contain a "main()" statement somewhere. All other routines are then linked into this "main()" routine, and execution of you program will begin here. On the Lattice C development environment, the public domain linker `blink' is used to link your program with the routines it needs from the standard libraries. On the Amiga, there are two major standard libraries which must be present to link even the simplest program. There is the library supplied by Commodore-Amiga called `amiga.lib' and the library supplied by the compiler manufacturer, in Lattice's case - `lc.lib'. amiga.lib is an interface to the routines of the Amiga's Rom Kernal. It allows your C program to call the assembly language routines of the kernal as if they were part of your program, providing linkages to the routines and the arguments they expect in the correct registers of the 68000. lc.lib on the other hand, provides the routines for the high level stream I/O and UNIX-like I/O, as well as functions for string, and other system manipulation. There are other libraries as well - for example the math libraries, which may be used by you program. Linkage is accomplished with a command similar to that for compiling, including a large number of options. Blink goes something like:

```
blink lib:c.o+xyzzy.o to xyzzy library lib:lc.lib+lib:amiga.lib
```

This is short! There are a dozen options which may or not be included, for example, whether to generate a MAP file or include debugging data or overlay info + this - that etc.... The above line will link our previously compiled program. The command translated into a variant of English says: link our program xyzzy but first include the code "c.o" which is the startup code. Then, search the libraries lc.lib and amiga.lib in that order, to find any routines you need to link into our program. Send the result to the file xyzzy.

Don't worry if this is all too much, we will go through it all step-by-step and in the group at the meeting.

A PLEA - If anyone knows a cheap place where we can teach C on machines in a group of about 10 over a few weeks or a place with enough room to fit 10 machines if we brought our own, then can they tell me about it - because until I find a place to set up some Amiga C evenings or afternoons, I will be drowned in requests. Do you have any idea how many people are phoning me about this??? If you want something done please put in a bit of effort and find us a venue, we'll do the rest!

Speed-up Amiga Floppies

by Glen McDiarmid

Think back to when you first started using KickStart 1.2. One of the differences between 1.1 and 1.2 that you may have noticed is the stepping rate of the floppy drives. The stepping rate is the rate at which the heads move across the disk when seeking to a track. KickStart 1.2 uses a faster stepping rate than 1.1, resulting in generally faster disk accesses.

The sequence of events when stepping the heads of a drive is this:

1. A 'step' signal is sent to the drive
2. The Amiga delays for a preset amount of time
3. Another 'step' signal is sent to the drive

On line two above, if the Amiga did not wait long enough for the drive to finish the previous 'step' command, the drive will not be ready to accept another step command, resulting in confusion as to which track the head is actually over. However, if this delay is longer than is actually required by that particular drive, it will take longer, generally, for disk access.

Therefore, the amount of delay that is required should be carefully selected, so as to have an efficient system. With KickStart 1.1, the stepping rate was much longer than was required on virtually all drives. The delay was decreased in KickStart 1.2, but it is still much longer than required for most drives. Will KickStart 1.3 set an even faster stepping rate? Frankly, I don't care, because I'm not waiting until then to make my drives work efficiently. Now that I know where in KickStart 1.2 the delay number is kept, I can set it to whatever I please, and it will be that way whenever I boot up.

Using the Mirror Hacker's package, I used SYSTEM TRACKS to load track 15, bottom side. On this track, I moved the cursor to offset $10C0. At this location, you should find '0BB8'. In decimal, this number is 3000. This number should be decreased, probably to somewhere between 1200 and 2000. It will take some experimenting on your part to find out just how fast your particular drives can step. I would suggest that you start off with a low number (fastest stepping rate), and if you get a read/write error when you boot the system, make the number somewhat larger, and try again. Make sure that your disk is write protected when doing tests. Remember, all drives must be able to keep up with the faster stepping rate. If you are testing DF0:, a good way to test is with "COPY DF0: TO NIL:".

I have found that $0700 (2000 decimal) is the optimum for my drives. If one of your drives is much faster than the others, you may independantly set the stepping rate of it using the public domain program "StepRate". In a test, "COPY DF0: TO NIL:" took 167 seconds at the normal stepping rate, and 160 seconds at the faster rate. While this may seem insignificant to some people, it is important to me because it saves me time not just once, but all the time from now on. Also, there is a bonus when you speed up the stepping rate - your drives will step with much less noise. Also, for those with egos that need a boost, you can tell others "my Amiga is faster than yours"!

[Editor's note: This article was reprinted from Issue 13 of the newsletter of the BAUG, the Brisbane Amiga Users Group.]

Amiga Users Groups

Here is (as far as we know) a complete list of Australian Amiga groups. If you are aware of any others, or you notice that some of these details are incorrect, please let us know. We are currently sending our newsletter to all these groups.

Canberra Amiga Users Society
36 Ambalindum Street
Hawker, ACT, 2614
BBS: (062) 59 1137

Australian Amiga User Association
PO Box 389
Penrith, New South Wales, 2750

Amiga Users Group
PO Box 48
Boronia, Victoria, 3155
BBS: (03) 792 3918

Brisbane Amiga User Group
PO Box 853
Toowong, Queensland, 4066

Northern Amiga User Group
PO Box 2457
Mount Isa, Queensland, 4825

Adelaide Amiga Users Group
GPO Box 332
Adelaide, South Australia, 5000

Amiga Users Group of South Australia
PO Box 486
Glenside, South Australia, 5064

Amiga Users of Northern Territory
c/o 4/4 Armidale Street
Stuart Park, Northern Territory, 5790

A Letter To AUG

from John Yiannis

George Wahr's letter raised some good points about the AUG BBS. If the cost of a second telephone line is too high, a possible alternative would be to do what Greg Hudson has done on his BBS. Speeds under 1200 bps are not supported (ie people cant use 300 bps) as down loading and up loading is just too slow and keeps the line occupied for too long. Also a faster modem that supports 2400 bps would save time as users with faster modems will then be able to upload and download in less time.

You mentioned that you have had a few complaints about the meetings being held in Clayton. How about a venue close to the city so there will be easy access to people in the northern and southern suburbs. Or you could alternate meetings, one month at Monash, and the next month it could be held somewhere like LaTrobe University. That way most users will then be able to attend at least 6 meetings a year.

Beginners Special Interest Group

by Pat Heuvel

The Beginners SIG convened for the first time at the November meeting. As was mentioned in Workbench, this SIG will run for a relatively short time after which I expect the members will have gotten tired of my simplistic rantings and moved on to greener pastures (What an appropriate appraisal...).

The aim of the SIG is to give the less experienced users of the Amiga a better understanding of this very special machine. To achieve this aim, I decided to use a hands on approach, so that anything I demonstrated, the audience could also do. This helps reinforce the subject, as well as keeping people interested.

The best format, I considered, was to discuss the machine from startup. When I first started up my machine, what I saw was Workbench. As I clicked on icons, other things came into view and life took on a whole new perspective (well, almost).

Subjects covered in this meeting centred around the Workbench and included:

* An introduction to Workbench
* Workbench windows
* Gadgets - Window to Back
  - Window to Front
  - Close
  - Sizing
  - Drag Bar
  - Scroll Bar
  - Disk Gauge
* A brief look at the Workbench Menu Bar
* Icons - Disk
  - Drawer
  - Tool
  - Project
  - Trashcan
* Preferences (and how to enable the CLI)
* A history of the famous Guru, presented by Peter Jetson

The time seemed to fly (maybe my own nerves) and before I knew it, it was time to hand over to the next bunch.

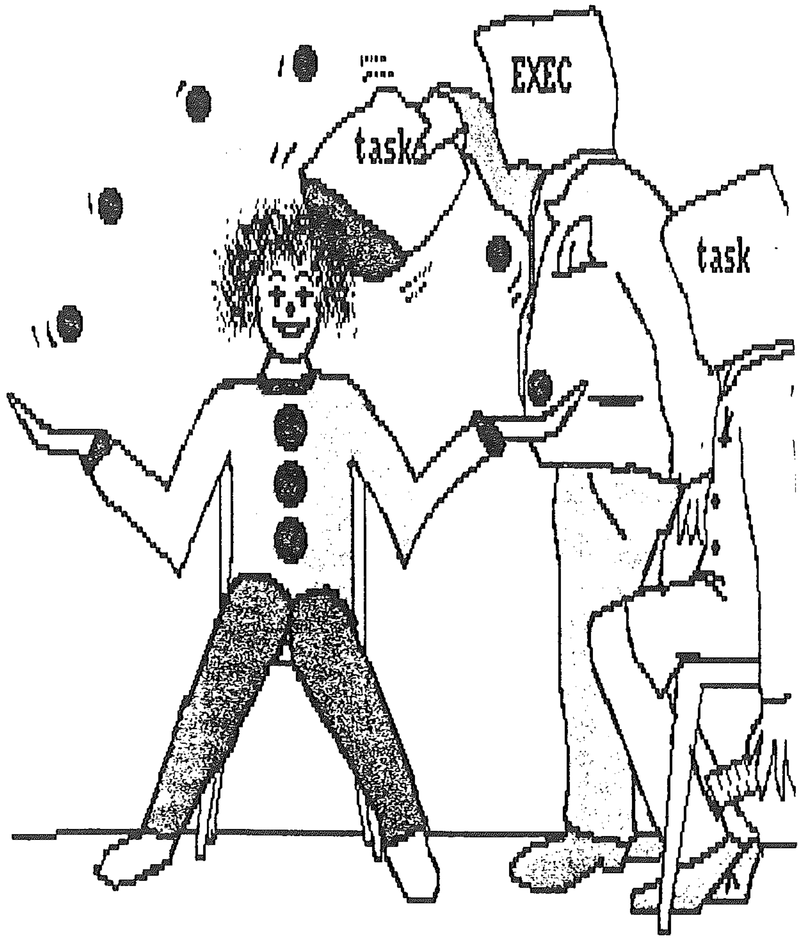
At the next meeting, I intend to cover some of the more common CLI commands, such as:

* Cd
* Copy
* Date
* Delete
* Dir
* Diskcopy
* Ed

This will give a grounding into the use of the CLI. I am aware that not many people use Ed these days, but it is common to anyone who has an Amiga, and it really isn't all that difficult to master.

Please note, I deliberately stayed away from machine specifics, since those people who own a 2000 or 500 should be familiar with setting them up and getting them going (the instruction book is good, at least for that much).

This concludes my summary of the Beginners SIG meeting, and for those of you who are interested, the rest of this article will be devoted to summing up the points raised in the meeting. In other words, if you don't want to read further, proceed to the next article...



Workbench and Menu Bar

Unless you have modified your startup sequence, chances are that your machine comes up running Workbench. Generally, when the startup has completed, the screen will be blank except for the Workbench title bar and one (or more) disk icons. Clicking and holding the right mouse button (referred to from now as the 'Menu' button) will cause the title bar to change, and display a number of words. These words are menu titles, and dragging the mouse pointer over one will highlight that title and cause a menu to pop up. Dragging the mouse down over the menu will cause each 'option' or selection to be highlighted while the pointer is in its vicinity. If the menu button is released while over an option (or the left button - the 'Select' button - is pressed while still holding the menu button), the operation represented by this option will come into effect. This is known as selection.

Icons

The word 'Icon' means 'Symbol'. On the Amiga, icons are used to represent a collection of information - be it a file for use with a drawing program, the drawing program itself, a disk, a trashcan or a drawer containing other icons. Each of these is a type of icon:

Disk — As the name suggests, this icon represents a disk. Clicking the select button twice over a disk icon causes a window to be opened, and (usually) more icons to appear.

Drawer — The default symbol for a drawer looks just like a drawer. As the name implies, it represents a container in which you may find other icons. As with the disk, double-clicking the select button usually reveals its contents.

Trashcan — This icon is exactly what its name suggests. If there is some object you wish to remove from a disk, you can move the pointer over that object, click and hold the select button, and drag that object over the trashcan. It will then disappear from the screen. From here, you have two choices: you may 'Empty the Trash' and permanently discard what is in the trashcan, or you may recall it. To empty the trash, click the menu button over the Workbench title bar, and select 'Empty trash' from the Disk menu. To recall, open the trashcan (like a drawer icon) and drag the required icon out. Note that you cannot delete a disk icon by dragging it over the trashcan.

Tool — This icon represents a program. Double clicking the select button over this icon will start the program.

Project — Most programs allow you to save data onto a disk. Many also create a special icon - called a project - for each file you create with the program. The project icon can be used to start up the program that created it, although there are certain exceptions to this (Textcraft 1.0 and 1.1, for example).

Preferences

Preferences is a tool designed to allow you to customise your working environment. With it, you can change the colours of the screen, the shape of the pointer, the distance the pointer travels in relation to the mouse; to name but a few.

Preferences can be started by double-clicking the select button over the preferences icon, or by typing 'sys:preferences' at a CLI prompt.

Most of the Preferences Gadgets are self-explanatory, so I won't waste space attempting a full explanation of all of them - besides, the Amiga manual (supplied with the machine) discusses preferences itself. Nevertheless, a summary follows (Please note that the description refers to Preferences as supplied with Workbench 1.2).

1. Preferences

I couldn't think of a better name for this screen. The first screen that appears on selecting preferences controls basic setup information. Without going to any of the other screens, you can change:

* Date and time
* Display text width (60 or 80 characters/line)
* Whether CLI icon is displayed in System drawer
* Screen colours
* Key repeat delay (delay before key starts repeating)
* Key repeat speed
* Screen positioning
* Whether the screen is interlaced (hi-res)
* Mouse pointer speed and
* Maximum delay for double-click on mouse

Before PJ jumps on me again, I will also mention that there are several other gadgets, which I have not yet discussed, leading to further preferences screens. From the above, it should be plain that a lot of workbench customising can be done right here. Nevertheless, on this screen, you will find gadgets enabling you to change:

* Serial setup
* Printer setup
* Mouse pointer

and last, but not least, there is a gadget to allow you to save your changes, one to let you cancel your changes and a further one to allow you to use the changes until the next time the machine is booted.

2. Change Serial

Here you can change the characteristics of your serial port for a printer, plotter, modem, etc. Note, however, that for these changes to mean anything, the program must refer to the preferences information (sys:devs/system-configuration). Refer to the accompanying documentation for details.

3. Change Printer

This screen allows you to tell the system what type

of printer you have, and how it is connected (Parallel or Serial port). It includes drivers for many popular printers, and many more are available from the public domain. It also allows you to specify pitch, quality and spacing (among other things). There is a caveat here, too: many word processors allow you to override these settings when you print documents. If you use '>prt:' from CLI (redirect to the AmigaDOS print device), the preferences settings will be used.

The Workbench 1.2 Enhancer booklet mentions narrow blank lines that may appear on graphic printouts when using Epson and similar matrix printers, and that this may be corrected by selecting the custom setting for paper size (just a little aside).

4. Graphic Select

The Graphic select screen determines how the Amiga will print graphic images. If you have a colour printer, the selection is easy (now someone is going to come along and prove me wrong, again) - Select color (silly Yank spelling) for the shade gadget and positive for the image gadget. For grey scale printing (like the shades of grey on a black and white photograph), select gray scale in the image gadget. Alternatively, selecting black and white here (as pointed out by PJ) will print some colours as black, and the rest as white; as determined by the threshold value. With the default setting of 2 for threshold, only the darkest colours will print as black; the rest, white. As the threshold number is increased, lighter colours will print as black. For more information, refer to the Amiga reference, chapter 7.

5. Change Pointer

This screen allows you to modify the pointer to whatever shape and colour takes your fancy. It includes four 'windows' of the basic workbench colours to show you what your new pointer will look like against each background colour. This is covered in more detail in Appendix 'C' of the Amiga reference.

Next Meeting

Having covered the basics of Workbench in the first meeting, I would like to move onto the directory concept and some of the more commonly used CLI commands, as an understanding of these two topics is a prerequisite for running some of the Public Domain software for which there are no icons (or which must be run from CLI by design).

Anyone in desperate need of help can ring me at home on 763-4548, and I will attempt to answer the query.

I hope I will be able to complete the next write-up in time for inclusion in Workbench (I'll try really hard). If enough people ask (and find my little summaries useful) I could possibly combine them into one document for separate publication at some stage. Time will tell. Meanwhile, enjoy...

==

Editor's Column

(Written 23-Jan-88)

Those who came had a great time at our January Barbeque, held in the wilds of Bundoora Park. The temperature calmed down a bit from the high thirties of the previous few days to the mid twenties. Just right for a BBQ.

It seems that most of the complaints about our new meeting time and place involve the time, or rather the change from Sunday to Saturday. The reason for the change was that the second Sunday was already booked for the year, so we selected the second Saturday. Considering the number of people who would rather meet on Sunday, we're going to check bookings for the first or third Sundays. Of course, a number of people dislike the extra distance they have to travel to get to Monash, just like a lesser number already disliked Burwood. The simple answer is that we can't find a suitable meeting place at a reasonable price anywhere near the centre of Melbourne. (The demographic centre of Melbourne is somewhere around Camberwell, by the way.) One possible solution (as a letter in this issue says) is to move right out the other side of town every second meeting. Your suggestions on this subject are welcome.

As mentioned elsewhere, I now have a 20 Mbyte hard drive attached to my Amiga, and it's fantastic! At the moment, I've only loaded my normal (full) CLI disk and a few other things like terminal programs, etc. In a few days, I'll load up Lattice C and all the other regular stuff I run. We're hoping to be able to get a good deal on the Wedge, stay tuned for details. In the meantime, you can see mine at the Feb meeting.

I expected to be running a Spirit internal 1.5 Mbyte board as well by now, but it still hasn't arrived. I ordered the board direct from the US, as a 0k board for US$299. That will work out at about $450 here, plus the extra for Customs (maybe as high as $120?) if they stop it. I've struck back luck lately, Customs have charged duty and sales tax on everything I've brought in, adding not inconsiderably to my costs. Still, I've been collecting and hoarding 256k RAM chips for a year or more now, so I've spread that part of the cost out. I've heard tales about a bug in the time and date routines that come with the Spirit board - this is the first leap year since the board was released, and the software doesn't like it. All dates until Feb 29th will be out by one day. Nice testing, guys....

Have you seen the new Amiga Logo?

News and Rumours

At the dealer show before Fall Comdex in the USA, Commodore officially acknowledged several long rumoured products, like a 68020 board for the 2000, a high-res monochrome display of about 800 by 1000 pixels and an 80286 BridgeBoard for the 2000.

At Comdex proper, there were several interesting products, like Microway's FlickerFixer, a de-interlacer for the Amiga 2000, selling for about US$595. The board eliminates interlace flicker by buffering the video image internally. It requires a MultiSync-type monitor for output.

The Commodore 68020 board runs at 14.2Mhz, has an on-board memory-management unit to oversee as much as 2 megs of 32-bit memory, and a socket for a 68881 arithmetic co-processor. No price was mentioned.

Also displayed was the "Video Toaster", a real-time, full colour digitiser and genlock, selling for about US$799.

Reportedly, Infinity Software was asked not to show Shakespeare (their desktop publishing program) as Comdex because Gold Disk was showing Professional Page. CSA was not allowed to show their 68020 products on the Commodore Stand because of Commodore's own 68020 board.

At AmiExpo, Comspec showed an autobooting hard disk for the Amiga 1000, using "magic" to boot within a few seconds of power-up, without a KickStart disk! ASDG showed prototypes of their Satellite Disk Processor, with its own 68000 and a half-meg of memory, and supports two ST506 hard disk drives and up to 56 SCSI devices!

Apparently, the new "newfs" file system for Amiga hard disks is still in Beta test stage, and will need a new KickStart and Workbench when released. It can boot from any device on the system, floppy, hard disk or network. Commodore is reportedly semi-officially working on version 1.3 of AmigaDOS, and is asking for suggestions on improving the operating system.

[Editor's note: Much of the above information is from Amazing Computing magazine.]

A Plea - Please Write Articles!

I'm a bit upset by the lack of newsletter submissions lately, they've meant a return to the four sheet newsletter. What this means is that our members (with a few notable exceptions) are taking more from our club than they are putting in. Almost all of you own Amigas, many of you buy software and hardware. Alas, few of you share your experiences with the rest of the club. All my interesting anecdotes, all my hardware and software purchased (few though they are!) have been written about in this newsletter. A few of our regular contributors are by far the busiest people I know, probably far busier than you or I. Unlike you, they find the time to pass on their experiences to everyone else.

It's quite likely that taken as a whole, members of the Amiga Users Group own every package available for the Amiga. Yet, each of us probably owns just a few, and would be interested in finding out about the programs we don't own. I started this group as a way of bringing all the information about the Amiga together, in one place, for the benefit of group members. Unless we spread our knowledge, we fail in what we aspire to be.

Please write newsletter articles about your experiences and purchases.

Public Domain Software Update

Again, we must bow thankfully to those Amiga programmers who have graciously allowed their works to be freely distributed. There are quite a few interesting programs in this month's releases, in fact I'm using one of them right now (the HandShake upgrade on Amigan disk #12).

Other things that caught my eye were the ARP collection, the hard disk backup utilities, the Space Invaders game, the Badge demo programs, the Toshiba printer driver and the V1.0 release of ConMan.

Thanks again to Fred Fish and A & J for their work in putting these collections together.

Fish Disk #119

MicroEMACS	Version 3.9e of Daniel Lawrence's variant of Dave Conroy's microemacs. This is an update to the version released on disk 93. Also included, for the first time, is extensive documentation in machine readable form. Includes source.

Fish Disk #120

Amoeba	This clone of Space Invaders is one of the best freely redistributable games for the Amiga to date. Unlike many commercial games, it even works correctly in a multitasking environment (by not requiring you to reboot just to play a game). Highly recommended! Binary only.
BackGammon	A graphical Backgammon game done as an undergraduate A.I. course project. Version 1.0, includes source.
Bankn	A complete checkbook system offered by the author as shareware. Version 1.3, binary only.
EgyptianRun	Cute little "road race + hazards" type game. Version 1.1, binary only, shareware, source available from author.
IconImage	Program to replace an old icon image with a new image, without affecting icontype, drawer data, etc. Includes source.

Fish Disk #121

BasicStrip	An AmigaBASIC program that helps to convert programs written in other forms of Basic to AmigaBASIC.
DataPlot	A shareware plotting program written in AmigaBASIC. Also includes a least squares curve fit program.
Plot	A shareware 3-D graphing program written in AmigaBASIC, with some sample output plots. Source available from author.
Stairs	This AmigaBASIC program demonstrates a musical illusion based upon perceptual circularity of widely spaced tones whose volumes are defined as a sinusoidal relationship to their frequency.
Uedit	Version 2.3 of this nice shareware editor. Has learn mode, a command language, menu customization, and other user configurability and customizability features. Binary only, shareware, update to version on disk 60.
WBColors	A simple little program to change the Workbench colors to a predetermined color set, for programs that expect to be booted off their distribution disk but instead are run from a hard disk. Includes source.

Fish Disk #122

Asteriods	Nicely done "blast the asteriods" type game. Unique feature is that all the images and sounds are replaceable by the end user. So instead of ships and rocks, you can have an Amiga against a horde of IBMPC's if you wish.
Iff2Pcs	An interactive puzzle program that takes any IFF file containing up to 16 colors, and breaks it up into squares to make a puzzle which the user can then piece back together again. Version 1.0, includes source.
Names	A shareware program to create and manage mailing lists. Binary only.
Pr	A little utility to print listings in different formats. Similar to the Unix "pr" program. Includes source.
PushOver	A neat little board strategy game, in AmigaBASIC. Push your pieces onto the board until you get five in a row in any direction. Includes source.
PuzzlePro	Create a puzzle from an IFF picture, which the user can then piece back together again. Written in AmigaBASIC. Version 1.0, binary only, shareware, source available from author.

Fish Disk #123

Arp	ARP stands for "AmigaDOS Replacement Project". Arp is an effort led by Charlie Heath of Microsmiths Inc., to replace the current DOS in a compatible fashion, so that current programs will continue to work. Arp also makes whatever improvements are possible, so that current and future programs will work better.
Car	This animation is one of Allen's entries to the Badge Killer Demo Contest. It apparently is an inside joke relating to a well known Amigan's experience with a certain highend graphics hardware manufacturer.

Fish Disk #124

Icons	Some sample animated icons. You might find just the icon for that refugee CLI program you've been meaning to make runable from the WorkBench environment.
Tarot	An AmigaBASIC program written by the author as an exercise for learning BASIC. Contains some nice graphic renditions of tarot cards.

Fish Disk #125

ElGato	This animation is Kevin's entry to the Badge Killer Demo Contest. It also has a background music arrangement, that requires Sonix to use.

Fish Disk #126

Colour	A program to manipulate the colors of specific named screens, saving their current color sets to data files, loading new color sets from data files, or interactively changing the colors. Includes source.
Dance	These two programs, "dancing polygons", are John's entry to the Badge Killer Demo Contest. They are variations of one another, but demonstrate the range of colors available on the Amiga. Includes source.
HBHill	This animation is one of Kevin's entries to the Badge Killer Demo Contest. It is the first known animation that makes use of the Amiga's "Extra Half Brite" mode.
Iconify	A subroutine that creates an icon on the Amiga screen that can be subsequently dragged around, and double-clicked on. You can use this to have your programs "iconify" themselves to temporarily get out of the user's way. Includes source and demo program.
OnlyAmiga	This animation is Iqbal's entry to the Badge Killer Demo Contest. It consists of tree balls being juggled by pyramids rotating on their tops.
Suplib	The support library needed to rebuild various programs of Matt's from the source, including DME, DTERM, etc. Includes source.
VCheck	Version 1.2 of the virus detection program from Commodore Amiga Technical Support. This version will test for the presence of a virus in memory, or on specific disks. Binary only.

Fish Disk #127

Bounce	This program is Steve and Tom's entry for the Badge Killer Demo Contest. It creates little dots that bounce

around and multiply. Includes source.

Nemesis This demo is Mark's entry to the Badge Killer Demo Contest. It is quite small for what it does, and won fifth place in the contest. Binary only.

Ripples This animation is one of Allen Hastings' entries to the Badge Killer Demo Contest. Unlike most other animations, it shows a fixed object from a moving point of view, rather than a moving object from a fixed point of view.

Fish Disk #128

Dis A 68000 disassembler, written in 68000 assembler. Includes source.

DropCloth DropCloth lets you place a pattern, a 2 bitplane IFF image or a combination of a pattern and image, into the WorkBench backdrop. Version 2.2, shareware, binary only.

LedClock An extremely simple clock program, for interlaced screens only. Includes source.

MRBackUp A hard disk backup utility, that does a file by file copy to standard AmigaDOS floppy disks. Includes an intuition interface and file compression. Version 1.3, includes source.

Paint A simple screen painting program, written in web. Requires web preprocessing program to rebuild from source. Includes source in web.

PrtDriver A printer driver for the Toshiba "3 in one" printer in its Qume (best) mode. Includes source in C and assembler.

SDBackUp A hard disk backup utility. CLI interface only. Does file compression. Version 1.1, binary only.

Sed A clone of the Unix sed (Stream EDitor) program. Includes source.

wKeys A "hot-keys" program that binds keyboard function keys to window manipulation functions (window activation, front to back, moving screens, etc). Includes source.

Amigan Disk #11

ConMan1.0 This is the release version 1.0 of ConMan, the brilliant console handler replacement by William S. Hawes.

DFree A handy utiltity for showing the amount of disk free space in BYTES, not blocks.

DiskMan2.0 Version 2.0 of DiskMan, which is a full screen directory utility using interlace and redefinable gadgets.

DoOp This is a tool for Workbench users which will give you the opportunity to vary your Startup scripts using gadgets.

ElGato You have to see to believe! A decent animated demo of a cat walking. Good to impress your non-Amiga friends with.

MemFree A utility to display the free memory in the Amiga. Very small and fast.

GOMF This is Get Out of My Face 1.0 by Christian Johnson. This catches the 'Software Error' gurus and attempts a recovery of resources. Shareware.

Ls Ls is a quick DIRectory command with some useful features.

Prefs Prefs replaces SetPrefs, which is a way to have several different preferences saved aside for certain situations. Also will just save or change your pointer.

uShow A very SMALL show program that shows them all, IFF, HAM, and even brushes!

WinSize Not only will this utility command set your window to some size, it can tell you the size of the current window!

Amigan Disk #12

AutoPics A quick and easy slideshow program. Will use YOUR favorite show program, or use one of the three included.

Handshake v1.41 Simply the most complete VT100 emulator available on the Amiga. New version. Supports XMODEM, YMODEM, and YMODEM batch transfers.

KickPlay A collection of files which give information on ways to modify your KickStart disk. (Not applicable to the 500 or 2000 machines.)

Movie A movie player, written by Eric Graham, author of Sculpt-3D, which is sold by Byte-by-Byte. Also includes utility files to make your own movies, and a sample movie file that will knock your socks off (Kahnanka).

Nemesis A demo written for the Zeroth BADGE Killer Demo Contest. Watch it now!

NewZap v3.1 An update to a file sector editor that has been floating around for a while. This new version includes keyboard shortcuts for all the commands.

QuickFlix Another slideshow player, can be used with virtually any file supported under the IFF standard.

SnipIt An amazing file which gives you a completely virtual clipboard to cut and paste 8 point text across any console window! (Early test version; if you have problems, send the author bug reports.)

WindowKeys A program which lets you cycle screens and windows without touching the mouse ... all keystrokes.

===

Newsletter Back Issue Order Form

Issue Numbers:	
Number of issues at $2 each	$
Less 10% for 5 or more	- $
Club Use Only	Total $
Mail to: Amiga Users Group, PO Box 48, Boronia, 3155	
Member's Name:	
Address:	

SOFTWARE ORDER FORM

Disk numbers :	
Disks supplied by Amiga User Group @ $8	$
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: ____________________

First name: ____________________ (no initials)

Address: ____________________

____________________ Postcode: ________

Phone Number: ________________ STD Code: ______

What services would you like AUG to provide: ______

____

____

Signed: ________________ Date: ________

Details on this side are optional

Year of birth: ________ Which model Amiga: ________

Occupation: ____________________

Interests: ____________________

____

____

Dealer's Name: ____________________

Dealer's Address: ____________________

____________________ Postcode: ________

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.

Club Use Only	Date	Paid	Rcpt #	Memb #	Card Sent

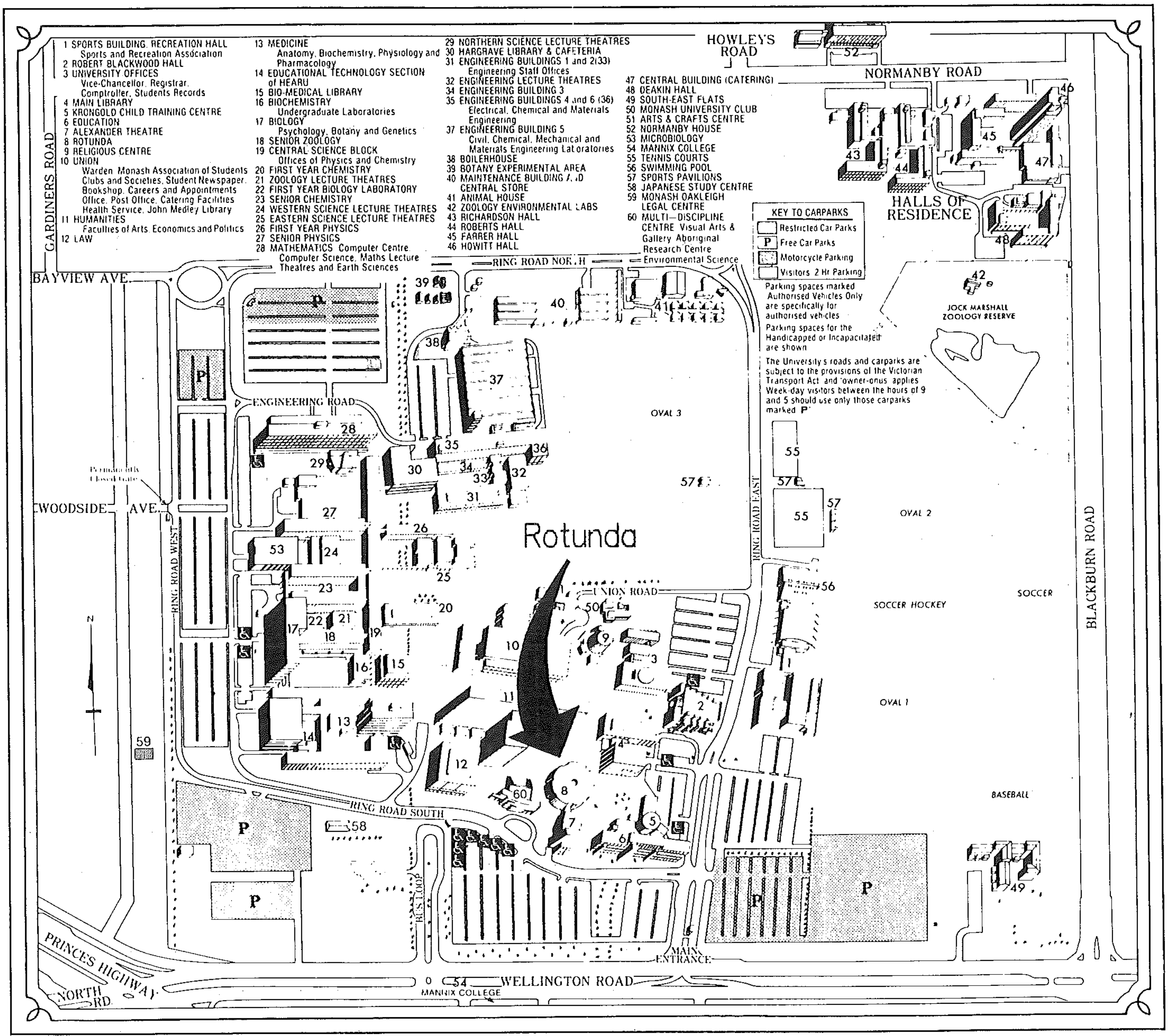
Amiga Workbench

Don't Forget

We now meet in the Rotunda, Monash University, Wellington Road, Clayton

The next meeting is at 2pm, Saturday, February 13th, 1988

Monash University is in Wellington Road, Clayton. See Melways Map 70, reference F10. Melways map 84A shows the University Campus in details. I've drawn a huge arrow on the map below to show where the Rotunda is. The best place to park your car is the car park area between Wellington Road and the Rotunda. The entrance to the Rotunda is virtually at the point of the arrow.



1 SPORTS BUILDING, RECREATION HALL
Sports and Recreation Association
2 ROBERT BLACKWOOD HALL
3 UNIVERSITY OFFICES
Vice-Chancellor, Registrar,
Comptroller, Students Records
4 MAIN LIBRARY
5 KRONGOLD CHILD TRAINING CENTRE
6 EDUCATION
7 ALEXANDER THEATRE
8 ROTUNDA
9 RELIGIOUS CENTRE
10 UNION
Warden Monash Association of Students
Clubs and Societies, Student Newspaper,
Bookshop, Careers and Appointments
Office, Post Office, Catering Facilities
Health Service, John Medley Library
11 HUMANITIES
Faculties of Arts, Economics and Politics
12 LAW
13 MEDICINE
Anatomy, Biochemistry, Physiology and Pharmacology
14 EDUCATIONAL TECHNOLOGY SECTION of HEARU
15 BIO-MEDICAL LIBRARY
16 BIOCHEMISTRY
Undergraduate Laboratories
17 BIOLOGY
Psychology, Botany and Genetics
18 SENIOR ZOOLOGY
19 CENTRAL SCIENCE BLOCK
Offices of Physics and Chemistry
20 FIRST YEAR CHEMISTRY
21 ZOOLOGY LECTURE THEATRES
22 FIRST YEAR BIOLOGY LABORATORY
23 SENIOR CHEMISTRY
24 WESTERN SCIENCE LECTURE THEATRES
25 EASTERN SCIENCE LECTURE THEATRES
26 FIRST YEAR PHYSICS
27 SENIOR PHYSICS
28 MATHEMATICS Computer Centre,
Computer Science, Maths Lecture
Theatres and Earth Sciences
29 NORTHERN SCIENCE LECTURE THEATRES
30 HARGRAVE LIBRARY & CAFETERIA
31 ENGINEERING BUILDINGS 1 and 2(33)
Engineering Staff Offices
32 ENGINEERING LECTURE THEATRES
34 ENGINEERING BUILDING 3
35 ENGINEERING BUILDINGS 4 and 6 (36)
Electrical, Chemical and Materials Engineering
37 ENGINEERING BUILDING 5
Civil, Chemical, Mechanical and
Materials Engineering Laboratories
38 BOILERHOUSE
39 BOTANY EXPERIMENTAL AREA
40 MAINTENANCE BUILDING AND
CENTRAL STORE
41 ANIMAL HOUSE
42 ZOOLOGY ENVIRONMENTAL LABS
43 RICHARDSON HALL
44 ROBERTS HALL
45 FARRER HALL
46 HOWITT HALL
47 CENTRAL BUILDING (CATERING)
48 DEAKIN HALL
49 SOUTH-EAST FLATS
50 MONASH UNIVERSITY CLUB
51 ARTS & CRAFTS CENTRE
52 NORMANBY HOUSE
53 MICROBIOLOGY
54 MANNIX COLLEGE
55 TENNIS COURTS
56 SWIMMING POOL
57 SPORTS PAVILIONS
58 JAPANESE STUDY CENTRE
59 MONASH OAKLEIGH LEGAL CENTRE
60 MULTI—DISCIPLINE CENTRE Visual Arts & Gallery Aboriginal Research Centre Environmental Science
KEY TO CARPARKS
□ Restricted Car Parks
P Free Car Parks
▨ Motorcycle Parking
□ Visitors 2 Hr Parking
Parking spaces marked Authorised Vehicles Only are specifically for authorised vehicles
Parking spaces for the Handicapped or Incapacitated are shown
The University's roads and carparks are subject to the provisions of the Victorian Transport Act and 'owner-onus' applies Week-day visitors between the hours of 9 and 5 should use only those carparks marked P
HOWLEYS ROAD
NORMANBY ROAD
HALLS OF RESIDENCE
JOCK MARSHALL ZOOLOGY RESERVE
GARDINERS ROAD
BAYVIEW AVE.
WOODSIDE AVE.
RING ROAD NORTH
RING ROAD WEST
RING ROAD EAST
RING ROAD SOUTH
ENGINEERING ROAD
UNION ROAD
BUS LOOP
MAIN ENTRANCE
WELLINGTON ROAD
MANNIX COLLEGE
PRINCES HIGHWAY
NORTH RD.
BLACKBURN ROAD
Rotunda
OVAL 3
OVAL 2
OVAL 1
SOCCER HOCKEY
SOCCER
BASEBALL
N
P

BY PUBLIC TRANSPORT . . . The simplest method is to take a train from Flinders Street or Loop stations on the Dandenong/Pakenham line to either Huntingdale or Clayton. Buses run from these stations to the campus or there is a taxi rank at Clayton. With suitable connections the trip takes about 45 minutes — but it can take longer! An inner neighborhood ticket will take you all the way via Huntingdale station and the bus, but you will need to purchase a comprehensive ticket for the trip via Clayton, which encompasses two neighborhoods. The campus is also served by buses from Box Hill, Blackburn, Belgrave, Chadstone, Jells Park-Glen Waverley, Dandenong-Mulgrave, Oakleigh and Elwood.

FROM THE CITY BY CAR . . . An easy route is along St Kilda Road or Kingsway/Queens Road and then on to Dandenong Road. The campus's tall Menzies Building comes into view a kilometre or so before the left turn into Wellington Road on which the main entrance is located. Allow 40-50 minutes for the trip. Drivers should note that restrictions apply in some car parks weekdays 9 a.m. to 5 p.m. and fines **do** apply. There is ample unrestricted parking and, closer to buildings, designated two hour visitor car parks — check the map or ask at the Gatehouse.

APAL for the AMIGA™

BY COMMODORE®

FULLY FEATURED 2 - DIMENSIONAL CAD

BASFORD SYSTEMS
25 Neale Street
Bendigo,
Victoria, 3550
Ph.(054) 43 3349

DRAWING FEATURES

Help
APAL is a command driven program and context sensitive help is provided via the Amiga Help key.

Lines
Line drawing commands follow a well structured syntax and provide over one thousand different ways to specify the geometry of a line. Special syntax is provided for Arrow head, Leader and Survey lines.

Curves
Like lines, many options are provided to fit circles and arcs to the required situation.

Text
Extremely flexible Text commands allow such things as any combination of horizontal and vertical justification of single as well as multi - line blocks of text. Multiple and user defined fonts are allowed in the one drawing.

Zoom
Comprehensive windowing options allow rapid redrawing and removal of drawing elements, as well as zoom in and out.

Texture and Thickness
Drawings may be done in up to nine different line thicknesses and up to six different textures. Texture patterns are user definable as well.

Layering
Drawings can be entirely on one layer or may be spread over up to thirty - two layers.

Cross hatching
Powerful easy to use cross hatching commands allow shading of simple and complex areas (including cut - outs) with user control over spacing, angle and starting point.

French curves
Smooth curves may be generated through a series of points.

Mouse digitising
The mouse may be used to digitise existing sketches and diagrams to any scale in either point or continuous mode.

Symbols
Any drawing can be used as a symbol and any drawing can be used in another drawing via scaling, rotation and translation options (others too). Transformations can be automatic by specifying two points, between which the incoming drawing must fit.

Startup file
Many drawing parameters take their initial value from the users startup control file. This allows the user to customise APAL to their needs.

Relocation
Parts of a drawing can be shifted around or copied while performing transformations such as scaling and rotation at the same time.

Grid
For drawings with much regularity, a grid can be used to "round off" drawing commands to the nearest grid point.

Deletion
Extensive commands are available for deleting individual drawing elements, such things as lines, curves,text, chains of lines, sub-pictures, etc. As well arbitrarily shaped areas of the drawing can be removed or isolated.

Amiga features
Just two of the many features of the Amiga supported by APAL are, user definition of the function keys and Amiga Menus.

Plotting
A weakness in many products but not APAL. Extensive options for plotting are provided. For instance lines can be thickened correctly, even on single pen plotters. Plotting by layers or thickness are just two other options.

Documentation and Support
APAL is provided on an Amiga standard 3 1/2 inch disc. It comes with a two hundred plus page manual and quick reference card. In the manual is a registration form so that users can be notified of updates (and there will be some) and tips. As the product is designed in Australia, local support is assured and feedback from users is encouraged directly to Basford Systems.

Applications
APAL is suitable for many draghting jobs from simple sketches to major technical works. Its major feature is its ease of use and hence productivity.

Compatibility
APAL is for the most part keystroke compatible with the leading minicomputer CAD system (VAX, Apollo), Palette. File interchange with Vax based Palette systems is even possible.

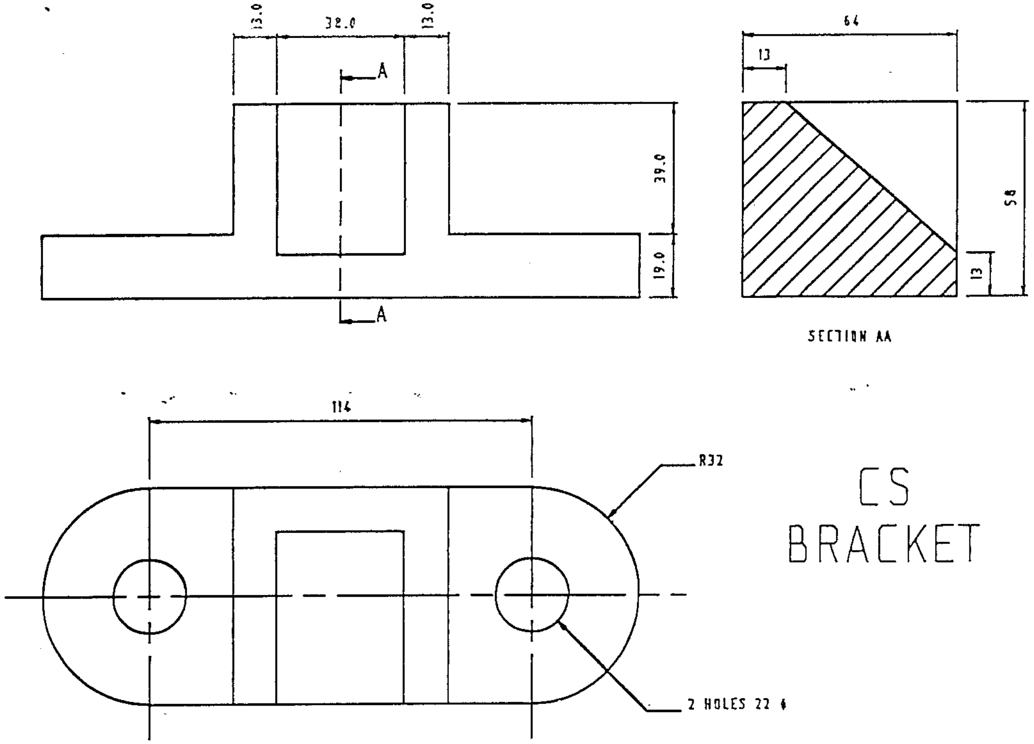
Requirements
APAL requires a Commodore Amiga model 500, 1000, or 2000. 512k is required although more can be used. AmigaDos 1.1 or 1.2 can be used with just the internal 3 1/2 inch drive.

Sample drawings
Several sample drawings are shown overleaf to illustrate the capabilities of APAL. Note that they are not to scale shown,but were originally and were simply scaled by the software for 1:1 output and reproduction here.

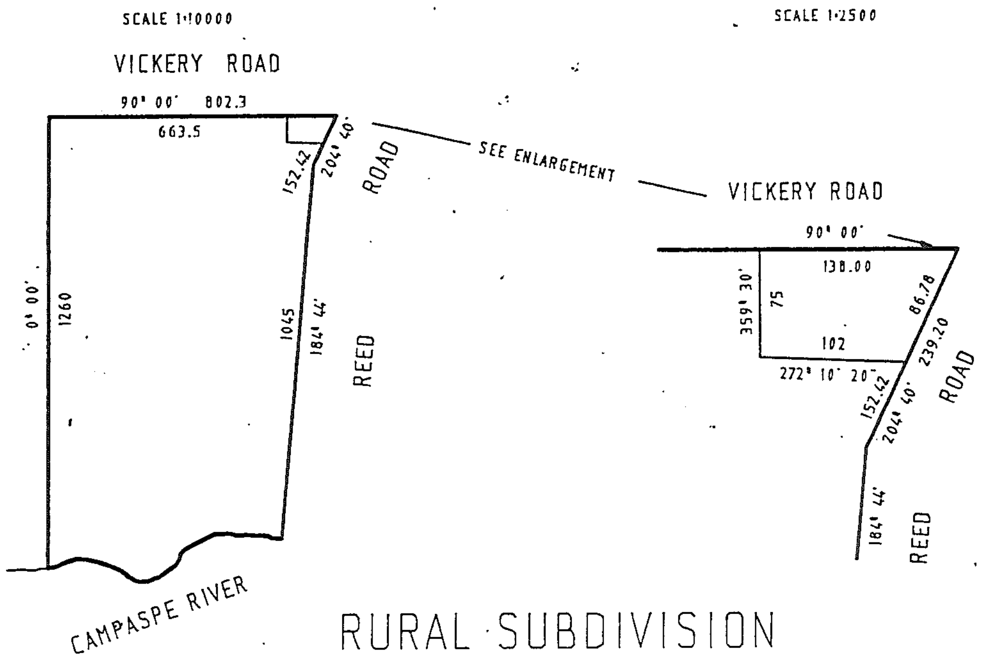
SPECIAL OFFER

$ 199.00

MADE IN AUSTRALIA

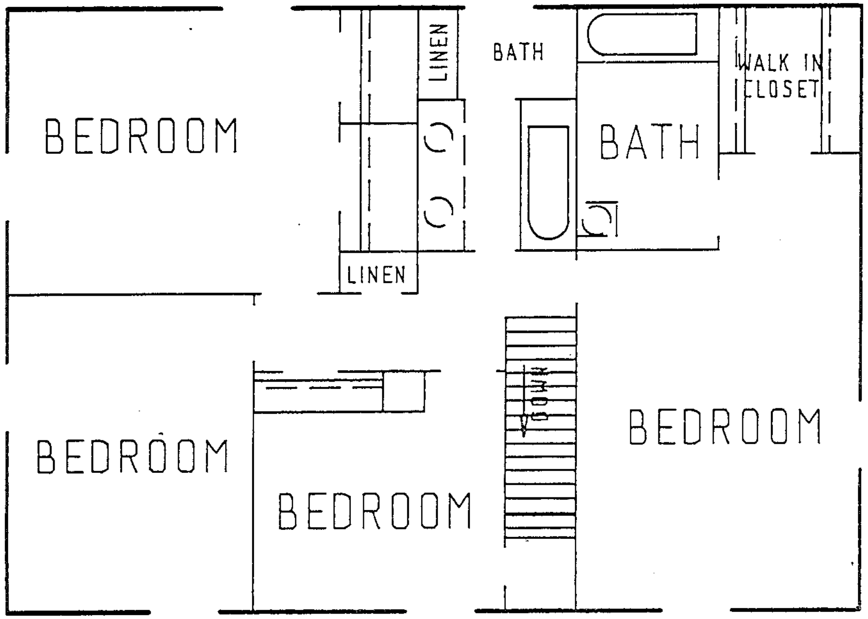


CS BRACKET

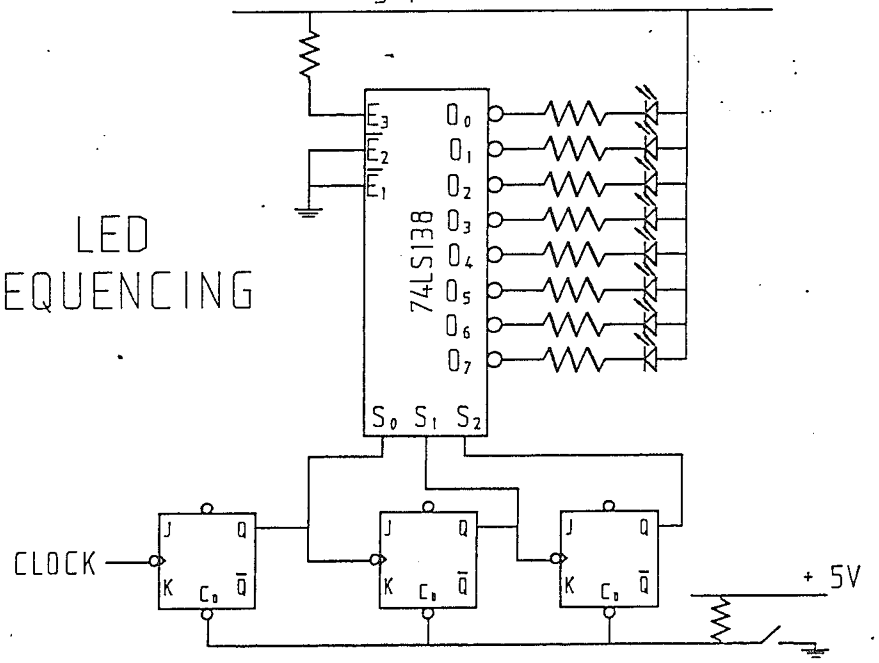


RURAL SUBDIVISION

UPPER FLOOR PLAN



LED SEQUENCING



ALL J K=1 AND S_D=1