

WIN
AVATEX MODEM

NEW MAGAZINE AND DISK

Amiga Times

News and Information Resource for the Commodore AMIGA™

Issue 1.1

US \$8.95

Canada \$9.95

Printed in Canada

Professional Page 1.1

Photon Paint

IntroCAD

Ferrari Formula I

FS II Scenery Disks

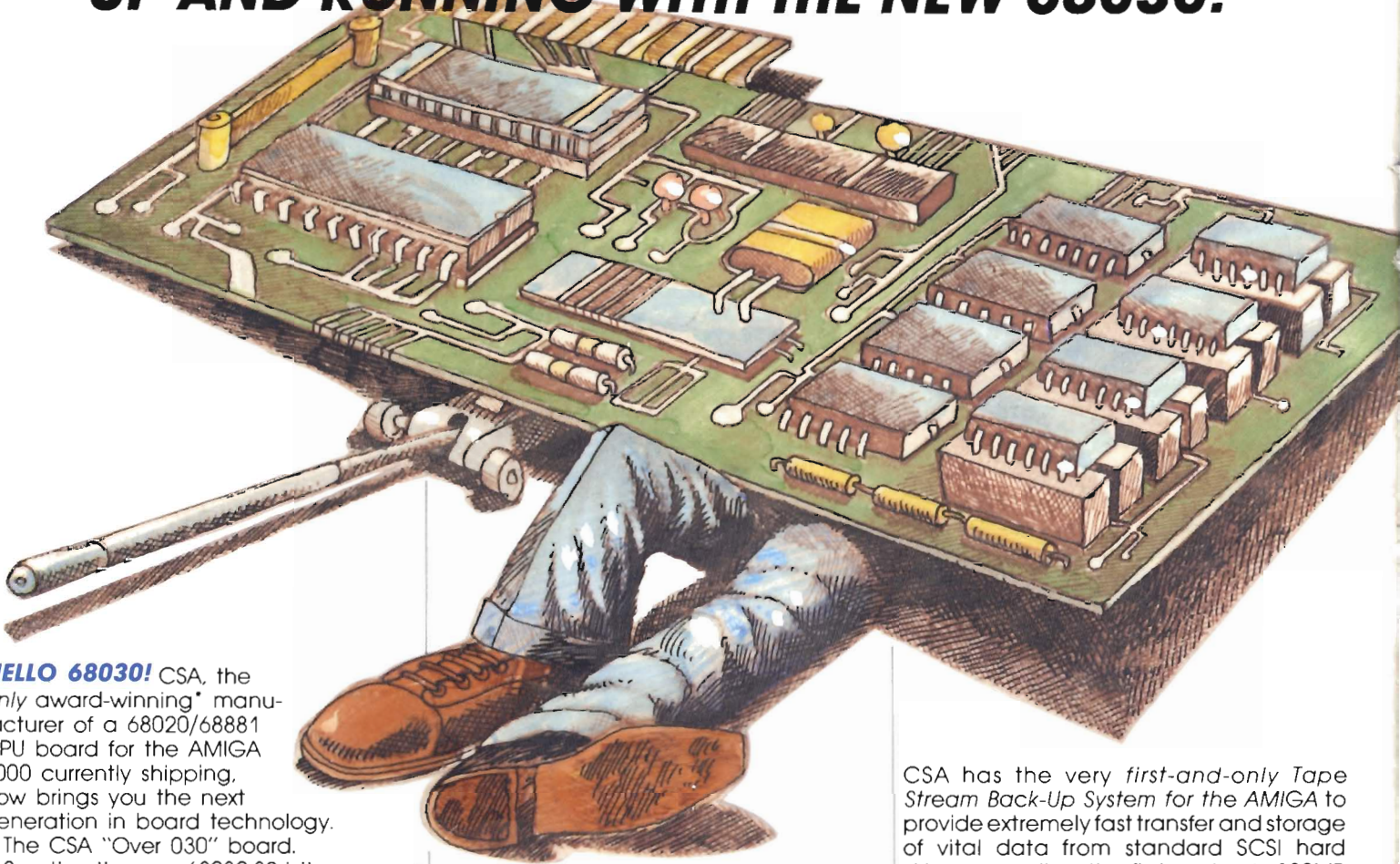
JET



**Assembler IFF routines
"C", Modula-2, Basic
Upgrade to 68010**

**Telecomm, MIDI
and more....**

WHILE THE OTHER GUYS ARE STILL TINKERING WITH THEIR 68020 BOARDS, CSA IS ALREADY UP AND RUNNING WITH THE NEW 68030!



HELLO 68030! CSA, the only award-winning* manufacturer of a 68020/68881 CPU board for the AMIGA 2000 currently shipping, now brings you the next generation in board technology.

The CSA "Over 030" board.

Sporting the new 68030 32-bit processor with onboard memory management, and 68882 math co-processor, Motorola reports that its new line of microprocessors are up to 4 times as fast as the 68020/68881 combination.

So, while the other guys are still promising 68020 boards for the AMIGA 2000 CPU slot, CSA is already in gear shipping out the new "Over 030" board for systems development nationally and internationally. We're already shipping the new 68882 math co-processor as an option for our current line of 68020 boards today.

THE BEGINNING:

To begin with, we know the 68020 and

68030 can access up to 4 gigabytes of RAM, and that you'll eventually need more memory expansion capability for running UNIX (when available), image processing, animation, CAD/CAM, and other "RAM Hungry" applications.

So CSA's advanced technology now brings you up to 32MB of 32-bit low cost dynamic RAM on a *single* board.

By the way, the other guys only give you up to 2MB of RAM on their boards, and that's all you get . . . forever.

THE MIDDLE:

In the middle of all this, we know that you'll need a way to protect your important programs from possible loss or damage. So

CSA has the very *first-and-only* Tape Stream Back-Up System for the AMIGA to provide extremely fast transfer and storage of vital data from standard SCSI hard drives, as well as the first-and-only 800MB WORM (Write Once, Read Many) Optical Disk Drive originally developed for the SDI "Star Wars" Program.

THE END:

All the new guys on the block are claiming they're "blowing us away" with superior boards . . . but in the end CSA is *still* the only one with independent benchmarks that show *superior* performance to the Macintosh II, VAX 8800, and Sun 3/160 Workstation.

The choice is *still* yours.

You can either wait for "vaporware" from the other guys, or you can call CSA for the real McCoy today.

CSA. We ship products, not promises.

The Technological Leader Stays 1-Step Ahead . . .



Computer System Associates, Inc.
7564 Trade Street, San Diego, CA 92121
(619) 566-3911 • TELEX 333-693

*Voted the Best Hardware Manufacturer for the AMIGA in 1986 and 1987. AMIGA is the trademark of Commodore Business Machines. Macintosh is the trademark of Apple Computer, Inc. VAX is the trademark of Digital Equipment Corporation.

AmigoTimes

Managing-Editor Eyo Sama

Marketing Director Lisa Sama

Art Director Olusegun Olaniyan

Data Manager Ernest Nagy

Assistant Editor Michael Hermann

Circulation Manager Alex Ribeiro

Contributing Authors Serge Boucher
John Foust
Michael Hermann
Gerald Hull
Harv Laser
Ernest Nagy
Olusegun Olaniyan
Eyo Sama
Mike Scalora

For

Advertising Sales
CALL
Lisa Sama
1-(514) 638-6304


Dealer Enquiries
CALL
1-(514) 638-6303

AmigoTimes® (ISSN 0836-0588) is published monthly by Sama Software Inc., 5124 St. Laurent, suite 100, Ville St. Catherine, P.O.B. 1228, Quebec J0L 1E0, CANADA. Subscription rates are US \$41.70 for 6 issues (in Canada CDN \$51.00), and US \$71.40 for 12 issues (in Canada CDN \$87.00). For Foreign Airmail, please inquire. Phone: (514) 638-6303. The entire contents of this publication are the copyright 1988 of Sama Software Inc. Even though *AmigoTimes* makes every effort to assure the accuracy of its contents, it assumes no responsibility for damages due to errors or omissions.

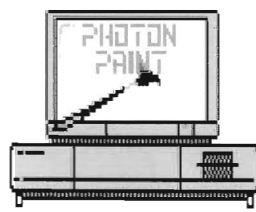
PRINTED IN CANADA

This issue of *AmigoTimes* was created entirely on Commodore Amiga computers with the aid of the following tools:


AmigoTimes



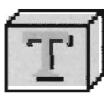
Professional Page 1.1




Photon Paint




Digi-View




TextCraft Plus




GRABIt




Deluxe Paint II




DrawPlus




GoldSpell 2.0



Butcher 2.0



ProWrite 2.0



PIXmate
Version 1.0

PIXmate

FEATURES

7 Manipulating IFF Pictures

by John Foust.

A look into what tools the Amiga artist has available for IFF image manipulations.

12 68000 / 68010 Should You Change?

by Olusegun Olaniyan

These two chips are compared against each other using a variety of benchmarks. Also provided are instructions to replacing an A1000's MC68000 with an MC68010.

24 Telecomm

by Harv Laser

Everything you need to know about Amiga telecommunications and modems, from connecting up to actually going online.

51 MIDI

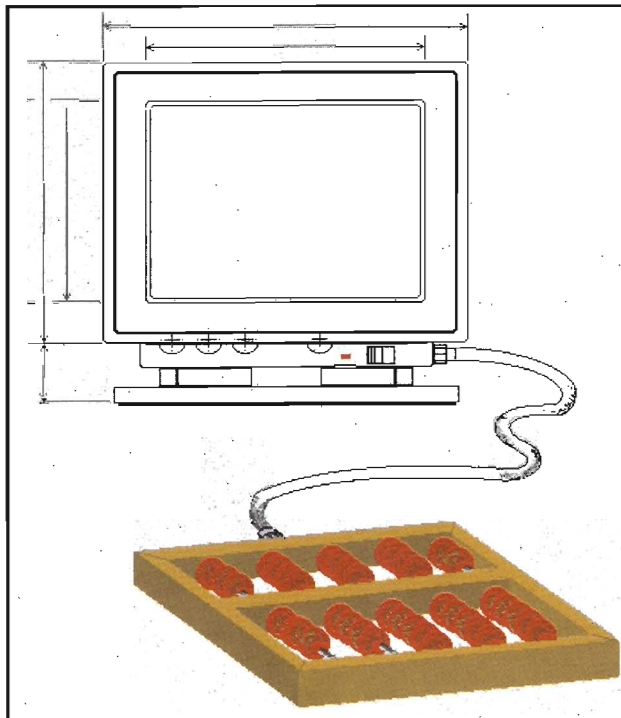
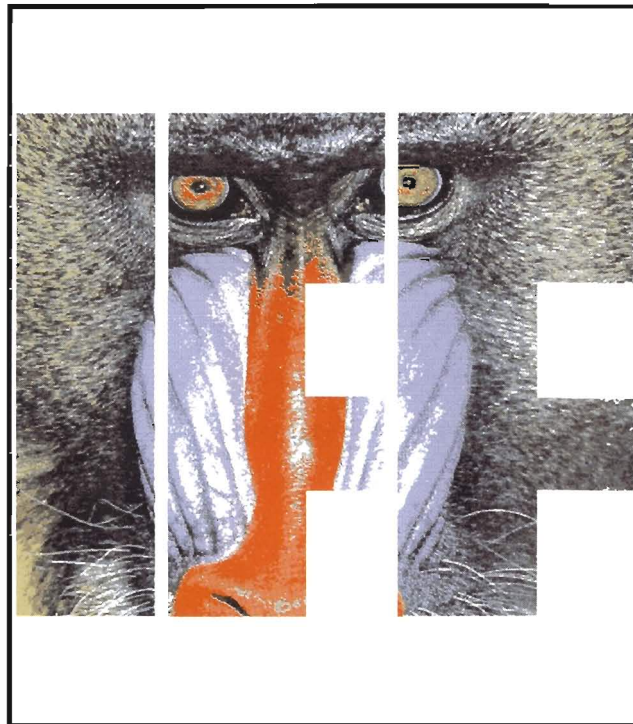
by Serge Boucher

A concise description of how computers have revolutionized the music industry with a multitude of midi applications.

52 The History of Computers (Part I)

by Ernest Nagy

The first of a two part in-depth article; this first part describes the foundation upon which the microcomputer was built.



PROGRAMMING

10 The Amigo Weatherman

by Michael Hermann

A program that applies the fundamentals of meteorology in the Basic language, and attempts to predict the weather by manipulating data that is entered by the user.

30 "C" Further

by Eyo Sama

Describes parameter passing, using the AmigaDOS Execute() function as your medium.

32 A Quicker Packer Upper

by Dr. Gerald Hull

This IFF image compression routine is written in assembler, and it turns out to be faster than EA's packer.c.

37 Modula-2 File Requester

by Mike Scalora

Helps guide novices through the Modula-2 world while going through the process of building a file requester.

R E V I E W S

17 **Ferrari Formula One**

by *Olusegun Olaniyan*

This is one of the most realistic, real time auto racing simulators; the only requirements are cat-like reflexes and nerves of steel.

20 **Flight Simulator Scenery Disks #7 & #11**

by *Michael Hermann*

Explore the east coast of North America with a little help from SubLOGIC's FSII scenery disks.

42 **Color Separation and Professional Page V1.1**

by *Eyo Sama*

Gold Disk has released their promised update to Professional Page. It has a built in color separator and dot matrix support, allowing even the novice to produce professional looking documents.

47 **Photon Paint**

by *Olusegun Olaniyan*

A new generation HAM paint program with the special effects capabilities of \$10,000 high-end graphics programs.

56 **IntroCAD**

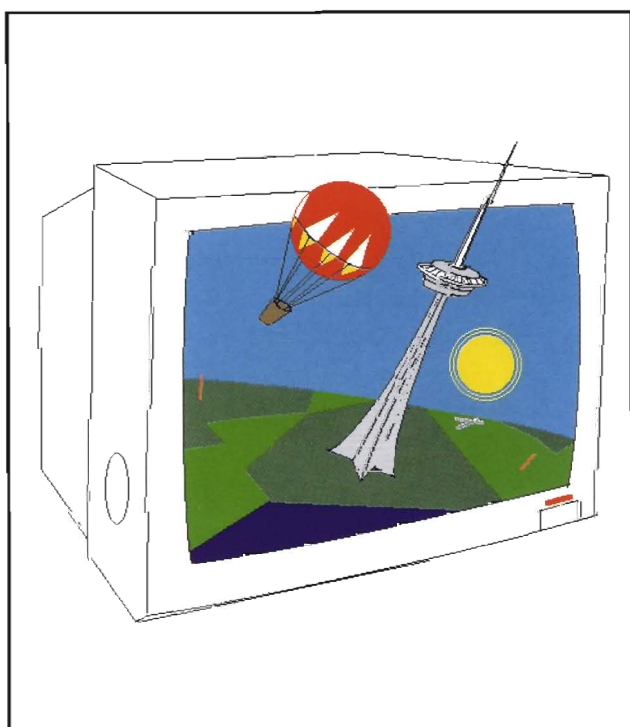
by *Harv Laser*

A new entry level CAD program, with a very good user interface, available at a great price.

58 **Jet**

by *Michael Hermann*

SubLOGIC has finally released JET, its popular fighter simulator in the tradition of Flight Simulator II, here is a hands on report.



D E P A R T M E N T S

4 **The Editor's Corner**

16 **Contest**

AmigoTimes "Amiga Image Contest"; many prizes to be won.

31 **Gobbledygook**

by *Ernest Nagy*

The drastic effects of over-work are seen in this column; we're not sure how this got by us, but maybe you can make sense of it.

61 **The Amiga Monitor**

Providing you with the latest breaking news and a synopsis of new products from third party software and hardware developers.

64 **Advertisers Index**



The Editor's Corner

It was three years ago, but it seems like just yesterday, when some of the staff members and myself stood wide-eyed and open-mouthed in front of the first Amiga to have appeared here in Montreal (it was displaying the famous Mandrill on its screen). We immediately knew that this was the machine for us, I'm sure many of you have had the same experience. At our local Amiga dealer we purchased the first few machines to come through the store's door, and since then, there has been no turning back.

Much has changed for the Amiga since those early days; we have seen the introduction of the Amiga 500 and the 2000, and we will soon see the introduction of the 2500AT, 2500UX, and the Amiga 3000 (for more details read the Amiga Monitor on page 61). We have gone through several AmigaDOS upgrades, each vastly improved from the last. Just yesterday I installed a new Kickstart 1.3 ROM into my 2000, this allows you to boot from the Hard Disk. It seems like a long time since we have seen any major releases from Commodore, but now there will be a flood of new products brewed up by the engineers and programmers from West Chester, Los Gatos, and Braunschweig. The fatter Agnus co-processor should be available soon, we actually saw this chip last week at the Amiga Developers Conference in Washington D.C., so it is not a myth anymore, it does exist. With this new chip you will be able to address 1MB of CHIP RAM as well as get other enhancements such as a 640x400 non-interlaced display.

We have seen the Amiga branch out into various fields of application such as Small Business, Graphic Design, CAD, Scientific Research, Music, DeskTop Video and most recently DeskTop Publishing, which of course brings us to our magazine, AmigoTimes. AmigoTimes is entirely created on Amigas and products created for the Amiga; we use various paint programs, digitizers, and other graphic tools to produce our images. Typesetting and the color separations are produced using Professional Page version 1.1. The capabilities and possible applications for our favorite computer seem endless, which is why we decided to publish this magazine on Amigas, for Amiga users, and by Amiga users.

With AmigoTimes we intend to keep you informed on the latest developments within the Amiga community. As you will see within these pages, we are trying to cover all aspects of Amiga computing. We will be bringing you reviews of the latest hardware and software as well as programming tips, tricks, and tutorials featuring several computer languages.

Each issue is accompanied by a 3.5" disk containing useful public domain programs and utilities, and you will never have to type-in a program listing from the magazine because the disk will contain both the source code and the executable files. The disk also gives you the opportunity to try-out demo versions of commercial software, which will aid you in the selection and purchase of any of the numerous packages now available for the Amiga. We strongly support the public domain and shareware concepts, therefore we are always interested in any programs you may have, the best of which will be included on the disk and also mentioned within the magazine. Also of great help would be comments and suggestions about the content and quality of both the magazine and the disk. If you give us an idea of some of the things you would like to see, or not see, in AmigoTimes, we will be in a better position to improve each issue.

If you are interested in winning a 1200 baud Avatex modem, a Ferrari Formula One game, or a

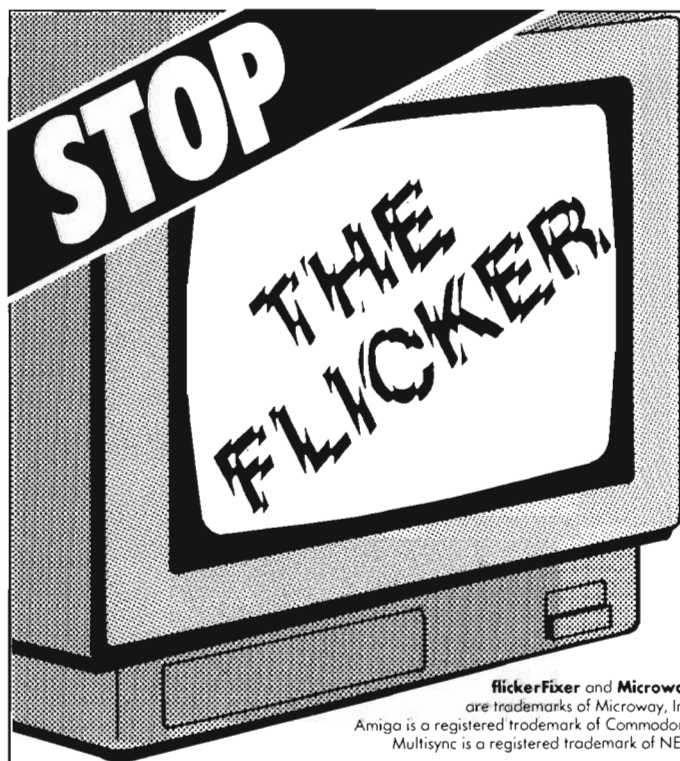
subscription to *AmigoTimes*, join the *AmigoTimes* Image Contest (see details on page 16). Just send us a disk with your original artwork and you will have a chance at receiving those prizes. In future issues we intend to have more such contests, some with other topics such as animation, music etc.

We are continuously receiving new products, of which we intend to give you thorough and honest reviews, except I am still trying to figure out how I can stop my staff from playing games like *ZOOM!* all day, it's hard to catch them, remember the Amiga is multi-tasking, all they have to do is pop the screen to the background. Actually, the truth is, we barely ever have time to play games, but when we do, I can be caught doing it just as often as they can.

The Amiga family is growing, there are now nearly 700,000 Amigas out there and the numbers are still increasing. We support the Amiga 500, 1000, 2000 and all future Amigas so we hope you will use *AmigoTimes* as a vehicle to grow with the Amiga. See you next month, I hope you enjoy the magazine.

I would like to dedicate our premiere issue to my parents, Sama and Traudl, and to the memory of Patricia Robitaille.

Eyo Sama
Managing Editor



flickerFixer eliminates your Amiga 2000's interlace flicker and visible scan lines. The result: **superior quality color or monochrome graphics and text** — for a full range of demanding applications, including CAD, desktop publishing, graphics, and video.

flickerFixer fits into the Amiga video slot, is fully compatible with all user software, and does not modify the standard Amiga video signals. The board also upgrades the Amiga 2000 with a flicker free 4096 color palette, has an overscan mode that features a screen size of 704x470 pixels and drives most of the popular PC EGA and VGA monitors, including the NEC Multisync and Mitsubishi XC1429C.

flickerFixer™

Advanced Graphics Adapter For The AMIGA® 2000

flickerFixer is priced at \$595. It is made in the USA by Microway, Inc. — "The World Leader in PC Numerics" since 1982. For more information or to order, call Microway Sales at (617) 746-7341 or your Amiga Dealer.

Microway

P.O. Box 79
Kingston, Mass. 02364 USA
(617) 746-7341

32 High St.,
Kingston-Upon-Thames, U.K.,
01-541-5466

58 Atchison St.,
St. Leonards, NSW, Australia
02-439-8400



DIGI VIEW

DELIVERS ULTIMATE GRAPHICS POWER

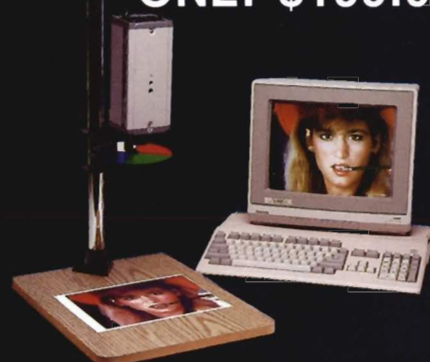
Bring the world into your Amiga with Digi-View, the 4096 color video digitizer. In seconds you can capture any photograph or object your video camera can see in full color and with clarity never before available on a home computer. Digi-View's advanced features include:

- Dithering routines give up to 100,000 apparent colors on screen
- NewTek's exclusive Enhanced Hold-and-Modify mode allows for exceptionally detailed images
- Digitize images in any number of colors from 2 to 4096
- Print, animate, transmit, store, or manipulate images with available IFF compatible programs
- Digitize in all Amiga resolution modes (320x200, 320x400, 640x200, 640x400)

"Digi-View sets new standards for graphics hardware" -InfoWorld

Digi-View is available now at your local Amiga dealer or call:
1-800-843-8934

ONLY \$199.95



NewTek
INCORPORATED

MANIPULATING



PICTURES

There are a half-dozen painting programs available for the Amiga. Why do so many exist? Each program offers features that another doesn't have, and a lot of people buy more than one paint program. An artist may need a special brush effect found in only one program. For someone who makes money with their Amiga, the extra cost of one more tool justifies a better result.

The differences between paint programs are often greater than brush effects. Deluxe Paint II doesn't understand HAM (hold-and-modify) images, so people bought Digi-Paint and Prism to modify HAM pictures.

Beyond having different features, each paint program understands several resolutions. The Amiga has low, medium, interlace and high resolution video. Low resolution has 320 by 200 dots, or

pixels, while high resolution has twice that, or 640 by 400. HAM images can be low and interlace resolution. The original Deluxe Paint didn't do interlace resolution. Most paint programs do not understand HAM.

These differences between programs lead to problems. How do you get from one resolution to another, to use one program for one effect, and another program for another?

For example, how can you move a HAM image into Deluxe Paint II? A HAM image may contain as many as 4096 colors, but Deluxe Paint II can only use 32 colors at most. If you try to load a HAM image into Deluxe Paint, you get an image with a very strange mish-mash of unnatural colors. Somehow, you need a way to convert the 4096 color image to one that only has

By John Foust

32 colors.

The artist mentioned before often encounters another problem in paint programs. Let's say they had made a 32 color picture of a city street scene. Later, they realize they could use a car from that picture in a new picture, so they try to clip the car as a brush from the first picture, save it, then load the second picture, and try to load the car - but no go. The first and second pictures have different colors in the palette, so when the car is loaded into the second picture, the colors change.

A new type of image manipulation program has come along to remedy these example problems. Two players in this market are Butcher II from Eagle Tree Software and PIXmate from Progressive Peripherals and Software. The primary function of these programs is not painting, but image processing. These programs specialize in converting IFF images from one resolution to another. They also have intelligent methods of changing the colors in a picture.

Before these image manipulation programs came along, some people used the Digi-View software to solve these two problems. The software that comes with the Digi-View video digitizer can reduce a HAM picture to 32 colors or less. It can also re-draw a picture using a palette from a second picture, so that when you move a brush from one picture to another, the colors match, and the brush is colored correctly.

To reduce a HAM picture to 32 colors, the image processing program determines which colors are most common in a picture. These colors become the fundamental colors in the palette of the new image. The program will also choose pairs of colors that look like a third color, in order to take advantage of a technique called "aliasing."

Aliasing is placing pixels of two different colors right next to each other. Your eye sees a third color, a blend of the two colors. In this way, the program can create the illusion of many more colors than are in the palette. Using these new, computed colors and mathematical techniques for matching colors, the program will recolor the old image with the new palette.

Aliasing works best when there are



This picture has been reduced to two colors, only pure black and pure white. Areas of solid color are now composed of aliased black and white, which your eye merges to look like grey. This image will be very clear if printed on a black-and-white dot matrix printer, or if imported into a desktop publishing program. The file size is 4684 bytes.

regions of color, instead of single pixels of random color. Aliasing needs a sequence of pixels in a row, so it can substitute two pixels of different colors for each pixel of the old color.

You can also reduce a picture to only two colors, white and black. Why would you want to do this? When you print a color image on a black-and-white printer, the results are often less than satisfactory. Shades of colors all become murky greys. If the image is converted to two colors, the results are often much better. Greys are now represented as aliased pixels of black and white, just like a newspaper picture.

If you use a desktop publishing program with a dot-matrix printer, you should use an image processing program to make "half-tone" pictures, such as a newspaper uses. Some programs such as PageSetter offer the ability to convert any colored image into black-and-white "half-tone" graphics for printing, but an image processing program will do a much better job of converting the picture to two colors than most desktop publishing programs.

To match the palette of one picture to another, the image processing program uses the same techniques as in converting a HAM image to less colors, except instead of computing an optimal

palette for the picture, it uses an existing palette from a picture. When one picture is re-computed with the palette of another, you can now clip brushes freely between the two images, with no color distortion.

Of course, if one image has all shades of blue, and the other all shades of red, there is little common ground between them. What does an image processing program do in this case? It does its best - meaning the results may not be what you expect. After all, you asked it to do the impossible. Color palette matching works best when the two images have a wide variety of colors, even though they may not be the same colors. With a variety of colors to choose from, the program can find "alias" colors that match the other picture.

Changing the resolution of an image without altering the image itself is another common problem. Low resolution has 320 by 200 pixels and interlace has 320 by 400 pixels, meaning that if you load an interlaced picture into a low resolution picture in Deluxe Paint, the image looks tall and stretched in the vertical direction.

One method of attack is to cut the picture into little square brushes, then resize the brush (using the halve or dou-



This is the same image as the original HAM below, reduced to eight colors. The file size is about half the HAM image, at 14016 bytes, but the image quality is still high. The picture is very recognizable.



This is the original image, a HAM picture with as many as 4096 colors. Like many pictures, it has somewhat less than that many colors, so converting it to less colors reduces the file size, but doesn't lower the image quality by the same amount. This file is 38410 bytes.

ble brush functions) and place the new brushes on a screen of the desired resolution. If you have only 512K of memory, then this method isn't the answer. It takes too long, and you probably don't have enough memory to do it in some cases.

Image processing programs handle this well, too. Changing an interlace picture to low resolution involves a loss of information. One way to convert a

320 by 400 image to 320 by 200 is to leave out every other scan line. The result has only half the information of the original. PIXMate can average each pair of scan lines, and make a new line from the two. This method tends to preserve some of the fine detail in an interlace image. There is no way to perform this averaging function with the brush resizing technique mentioned above.

As you learn more and more about

image processing programs and the formats of IFF pictures on the Amiga, other fun conversions are possible. By reducing a picture to four colors, and making those four colors the color of your Workbench, and using a public domain program that converts brushes to icons, you can convert full-size images to icons. Now the icons for your pictures will look like a tiny version of the picture.

Or you can use a public domain program called DropCloth to put a four-color image on the Workbench screen. Your icons and Workbench windows will show up on top of the backdrop of the IFF picture. Right now, I have a picture of David Letterman on my Workbench screen. □

Butcher II

Eagle Tree Software

PO Box 164

Hopewell, VA 23860

(804) 452-0623

\$37.00

Digi-Paint

Digi-View 2.0

New Tek

115 West Crane Street

Topeka, KS 66603

(913) 354-9332

\$59.95

\$199.95

Deluxe Paint II

Electronic Arts

1820 Gateway Drive

San Mateo, CA 94404

(415) 571-7171

\$129.95

PIXmate

Progressive Peripherals & Software

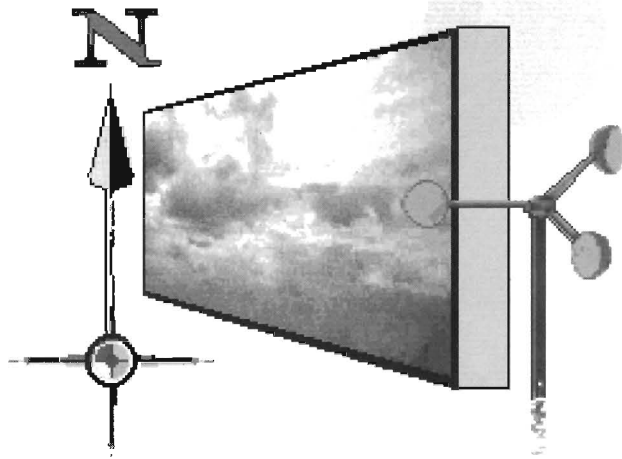
464 Kalamath Street

Denver, CO 80204

(303) 825-4144

\$49.95

The Amigo Weatherman



Forecasting the weather is not as complicated as the big weather stations would have you believe. All that's required is some understanding of how the weather system operates and a couple of instruments. Together with the BASIC program on the disk, you're now ready to forecast the weather with the best of them.

METEOROLOGY 101

I implemented fairly simple principles for this program. Wind and barometer readings can give a good indication of future weather conditions. The barometer measures atmospheric (air) pressure. However, the actual level of air pressure doesn't tell you very much unless you also know at what rate it is changing and if it is falling (decreasing) or rising. Next, you will need to know the direction of the wind.

Two types of major pressure systems exist in our atmosphere, lows and highs. A low pressure system brings with it rain, clouds, storms and generally bad weather. High pressure systems bring little wind, clear skies and dry air. A low pressure system moves in a counter-clockwise direction while a high pressure system turns in a clockwise direction. According to the Buys-

Ballot law, if you stand with your back to the wind, there is a low pressure system to your left. If the barometer is falling, that low is headed your way.

INSTRUMENTS

In order to obtain the necessary data, first and foremost you must have a barometer. If you don't own one you might consider buying one from a hardware store since they are usually inexpensive. You will also need a way to determine the present wind direction. If you have a wind vane, great! If not, it's not hard to make one of your own. Just get a long pole of wood (for height), a pointer (usually shaped as a wide arrow) with a nail through its center in order for it to spin around. But if that is too much for you, use a compass and observe the cloud or tree movements. Flags are okay, too.

USING THE PROGRAM

To start the program, copy the icon, named "Amigo Weatherman", from the enclosed AmigoTimes disk to your AmigaBasic disk and double-click on the icon. Or from Amiga Basic, type: RUN "Amigo Weatherman". Your screen will now show three windows. Click on the menubar at the top of the

screen and pull down the "ENTER DATA" menu. Select "Barometer - Present" to enter the current barometric pressure. Then click once in the window marked ENTER DATA to ready it for input. There are three measurement types available: Millibars, Inches, and Centimeters. Choose the measurement that your barometer uses. After that, enter the barometric pressure. The window should clear and the new barometric value will appear in the READINGS window.

Following that is "Barometer - Status". Select it to enter the direction the barometer level is moving towards. You'll also need to know how fast it is moving. To do this, write down the barometer reading about an hour before you run the program. If you are using millibars, it is considered moving rapidly if the pressure changes more than 2.03 mb in one hour. Now, select the "Wind Direction" menu. Remember that the direction (compass heading) you enter is the direction where the wind is *coming* from.

Once you've entered all of the data, a new menu will appear. Click on the top of the screen and under "FORECAST", pull down to "General Weather". The program will now use the information that appears in the READINGS window and will formulate a forecast. The forecast message will appear in the FORECAST window. If the information that you entered is incorrect, you may get a message in the FORECAST window asking you to recheck your data.

CLOSING NOTE

When I wrote the program I tried to keep it as simple as possible so the beginners at AmigaBasic could understand how it worked. I'm sure that with a little tinkering, if you add more inputs (such as cloud types and wind speed)

By Michael Hermann

you could improve on the program's forecast accuracy.

Remember, you do not need to type in this program, for your convenience, and to avoid the headaches usually associated with debugging, we have included it on the accompanying disk.

THE AMIGO WEATHERMAN

```

*****
** ... Amigo Weatherman ... **
** ..... by Michael Hermann ..... **
** AmigoTimes Public Domain **
*****

ON BREAK GOSUB Quit
BREAK ON
GOTO SETUP

Active:
M=MENU(0)
CHOICE = MENU (1)
ON M GOTO ProjectMenu,EnterMenu,ForecastMenu
GOTO Active

ProjectMenu:
ON CHOICE GOSUB About,Quit : GOTO Active

EnterMenu:
ON CHOICE GOSUB BaroPresent,BaroStat,WindDir
IF BarPres + BarStat + WdDir = 3 THEN
  BarPres = 2
  GOSUB ThirdMenu
END IF
GOTO Active

ForecastMenu:
ON CHOICE GOSUB GeneralWeather
GOTO Active

Keypress:
KPS = INKEY$: IF KPS="" THEN GOTO Keypress
RETURN

BaroPresent:
WINDOW 4
CLS
PRINT
PRINT " Barometer - Present reading"
PRINT " -----"
PRINT
IF bm=0 AND bm<4 THEN GOTO BarValueEntry
PRINT " [1] millibars"
PRINT " [2] inches"
PRINT " [3] centimeters"
PRINT
GOSUB Keypress
IF KPS="1" OR KPS="2" OR KPS="3" THEN GOTO BaroPresent
bm = VAL (KPS)
GOSUB ReadingsDisplay
WINDOW 4

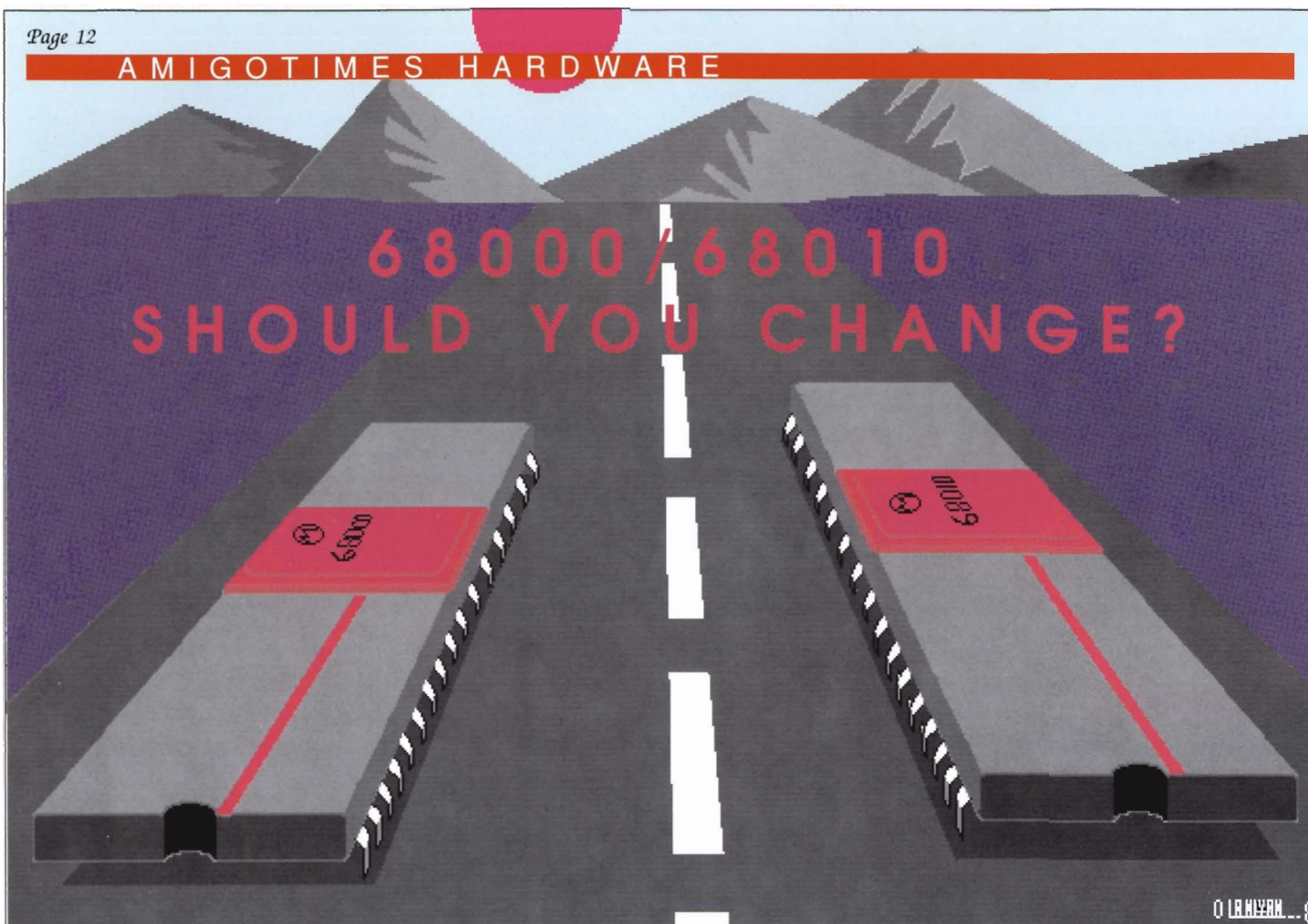
BarValueEntry:
PRINT
PRINT " What is the present"
PRINT " reading? :";bp
IF bm=1 AND (bp=>897.4 AND bp<1117.5) THEN
  CLS : GOTO BarPresentFin
END IF
IF bm=2 AND (bp=>26.5 AND bp<33) THEN
  CLS : GOTO BarPresentFin
END IF
IF bm=3 AND (bp=>67.31 AND bp<83.82) THEN
  CLS : GOTO BarPresentFin
END IF
CLS : PRINT
PRINT " ** value out of range **"
PRINT
PRINT " Range allowed: "
PRINT " 897.4 - 1117.5 mb."
PRINT " 26.50 - 33.00 in."
PRINT " 67.31 - 83.82 cm."
PRINT
FOR delay = 1 TO 1500 : NEXT
GOTO BarValueEntry

BarPresentFin:
BarPres = 1 : REM ** represents selected
rd0 = bp
rdcc = 20
IF bm=1 THEN rdcc = 10
GOSUB RoundOff
rbp = rd0
GOSUB ReadingsDisplay
GOTO Active

RoundOff:
rd1 = rd0 - INT (rd0)
rd1 = rd1 * rdcc
rd1 = INT (rd1 * .5)
rd1 = rd1/rdcc
rd0 = INT (rd0) + rd1
RETURN

BaroStat:
WINDOW 4
CLS
IF BarPres=0 THEN
  PRINT " Please select: "
  PRINT
  PRINT " ** Barometer - Present **"
  FOR delay = 1 TO 5000
    NEXT

```



The Motorola MC68000 (the first member of the M68000 family) is the heart of every Amiga 500, 1000, and 2000. Many manufacturers have joined Motorola in sourcing M68000 chips. A number of these manufacturers, including Commodore Amiga, are expanding the development rate of VLSI (Very Large Scale Integration) peripheral chips to support the M68000 family, and thereby making it possible, in certain cases, to have mainframe performance in a desktop personal computer. In the case of Commodore, the peripheral chips in question are Agnus, Paula, and Daphne; these three "ladies" have been designed to work very well with the MC68000 and are the core of the Amiga's multi-tasking capabilities. The M68000's 32-bit architecture, and its large linear address space provide an ideal environment for UNIX (an operating system

developed by AT&T which gave rise to programs targeted at the VAX and other powerful computers).

Originally, this article was not written with the intention of benchmarking the MC68000/68010 microprocessors, but rather as a cheap hardware hack intended to coax just "a wee bit more speed" out of Amy. The idea of exchanging the CPU began a while back when I received a Product Preview information sheet from Motorola Semiconductors regarding the MC68010. The preview sheet indicated that the MC68000 was the basis for future processors and that each of these processors would retain compatibility with its predecessors. This explains why there isn't too much of a compatibility problem with 68020 and 68030 boards. The Motorola 68010L8 microprocessor shares pin for pin compatibility with the Amiga's native Motorola 68000L8, and

the added advantage of an increased system performance.

To machine code programmers, the MC68010 has two extra instruction sets (machine code commands) as well as improvements on those already existing within the MC68000. Along with improved instruction sets are improved execution times for these instruction sets. The execution of 32-bit arithmetic and logic data operations in registers on the MC68010 will generally be two clock cycles shorter than they are in the MC68000. This can translate to 25% better performance for these types of operations, this is our area of interest. After reading further into the Product Preview sheet I was able to discern that a problem may arise from an exchange of the CPU'S and that this problem could occur at the user level (i.e. the coding of software). I will refer back to

(continued on page 14)

By Olusegun Olaniyan

Subscribe to . . .

AmigoTimes

News and Information Resource for the Commodore AMIGA™

AmigoTimes Reader Poll

1. What type of machine do you own?
 - ☐ Amiga 500
 - ☐ Amiga 1000
 - ☐ Amiga 2000
2. Where do you use your Amiga?
 - ☐ At Work
 - ☐ At School
 - ☐ At Home for business
 - ☐ At Home for recreation
3. What peripherals do you own?
 - ☐ Printer
 - ☐ Hard Disk Drive
 - ☐ Modem
 - ☐ Memory
 - ☐ Sidecar
 - ☐ Genlock or Digitizer
 - ☐ Musical Equipment
 - ☐ Other Specify
4. What are your interests?
 - ☐ CAD
 - ☐ Music
 - ☐ Desktop Publishing
 - ☐ Desktop Video
 - ☐ Graphics
 - ☐ Games
 - ☐ Other Specify
5. Do you find AmigoTimes...
 - ☐ Complex
 - ☐ Biased
 - ☐ Dull
 - ☐ Interesting
 - ☐ Great
6. Are you a subscriber to AmigoTimes?
 - ☐ Yes
 - ☐ No
7. What other Amiga Magazines do you read.
 - ☐ AmigaWorld
 - ☐ Amazing Computing
 - ☐ Amiga Transactor
 - ☐ Amiga Sentry
 - ☐ Robo City News
 - ☐ Info
 - ☐ Byte
 - ☐ Other Specify
8. What topics would you like to see in future issues of AmigoTimes?
 - ☐ More Programming
 - ☐ More reviews
 - ☐ More features
 - ☐ Applications for the Amiga
 - ☐ Hardware projects
 - ☐ Desktop Video
 - ☐ Music
 - ☐ Other Specify
9. How would you rate this issue of AmigoTimes?
 - ☐ Terrible
 - ☐ Poor
 - ☐ Fair
 - ☐ Good
 - ☐ Very good
 - ☐ Great

Send Check or Money Order
Payable to:

AMIGOTIMES

5124 St. Laurent, Suite 100
Ville Ste. Catherine, POB 1228
Quebec, J0L 1E0,
CANADA
tel. (514) 638-6303

FOR THE AMIGA EXPERIENCE . . .

Yes! **BEGIN** my subscription to **AmigoTimes** for:

- ☐ Six months (6 issues) at US \$6.95 (CDN \$8.50) per issue US\$41.70 (CDN \$51.00)
- ☐ One year (12 issues) at US \$5.95 (CDN \$7.25) per issue US \$71.40 (CDN \$87.00)

TO:

MR./MRS./MS.

Please Print full name

COMPANY

ADDRESS

CITY

STATE/PROV.

POSTAL CODE/ZIP

Please allow 30 to 60 days for delivery of first issue.

(continued from page 12)

this later on in the article.

Despite the potential problem that may arise I decided to proceed. One of the attractions of this project was the ease with which it could be accomplished, the low cost (\$32.00 CAN), and the minimal risk involved.

THE OPERATION

Upon obtaining a 68010L8 CPU, I proceeded to install it carefully. Although I will include step by step instructions on how to go about opening up your Amiga A1000 and exchanging your CPU, I cannot stress enough the importance of the Disclaimer at the end of this article; the changing of the CPU will invariably void your Commodore Warranty. Should you now be absolutely petrified after reading those last few words, yet still have a desire to "test-drive" the economy version of the "Turbo Amiga", seek out the help of "someone that knows something about computers" to help you do the installation. Now with all prophecies of doom and gloom aside, let us forge on.

A great deal of care should be taken during the opening of the Amiga, should you decide to attempt the replacement by yourself, the tools required are as follows: One Phillips head screw-driver and a small flat-head screwdriver.

1. Step one, disconnect all cables and connectors attached to the main unit and lay it on its back. Now in order to gain access to the interior of the Amiga, it is first necessary to remove 5 recessed phillips head screws underneath the main unit; after removing the recessed screws cover the holes with tape so as to prevent the screws from falling out and being lost.
2. After taping the recesses, turn the main unit right-side up and with either strong fingernails or with your flathead screw-driver gently pry the lid off the main unit.
3. Lifting the lid off you will first observe a rather interesting surprise on the underside of the lid. You will see the Amiga RF shield (a metal casing that covers the main board), this must be removed. In order to remove the shield you must untwist a few metal tabs that serve as anchors. After gently untwisting the tabs you will have to remove roughly 15 screws which hold the shield in place. The shield should lift off quite easily; if it does not, check to ensure

that all the restraining screws have been removed (DO NOT FORCE THE SHIELD OFF).

4. With the shield off you are now able to view the Amiga in all its glory. Looking from above with the main unit facing you, you can now see the 68000 CPU located on the right hand side at the end of the disk drive.
5. Now comes the critical part of this entire project, very gently with the flathead screwdriver, pry the 68000 up from its socket taking care not to scratch the PC-board that the socket is mounted on; note that the 68000 might resist removal, try to be gentle yet firm. When you have removed the old CPU, place the new 68010 into the socket and ensure that the pins are perfectly aligned, then gently press the new CPU into place (NOTE - this may take a small amount of force to press the CPU into place but use discretion).
6. After completion of the last step you are now home free, replace the RF shield, install the screws, replace the lid and replace the five main screws. Attach all connectors and power-up. If the computer does not boot-up don't panic, if you were careful and you followed instructions, then chances are that your 68010 CPU is dead; retrace your steps and replace the old 68000 and re-boot. At this point your kick-start requester should return, if it does you definitely have a dead 68010 CPU, on the other hand if nothing happens start to worry because you may have accidentally disrupted the internal workings of your Amiga. Should you have damaged something seek the assistance of someone that is fairly knowledgeable in electronics or a Commodore service center. However, having opened my Amiga on more than one occasion, I have found that if you're careful the chances of damaging something is low.

The first program to be tested with the newly installed 68010 was DrawPlus TM by Aegis Development. Having used DrawPlus on numerous occasions, any "change in system speed" should be noticed. The first noticeable change was in the redrawing of the window after a re-sizing, or after using the zoom-in/zoom-out feature. The next point of interest were block filling routines, like those that are used for filled squares and circles. The increased rate seemed evident, but to be absolutely certain a more qualitative test was needed. Benchmarks now became the next logical point, and in the days that followed I proceeded to collect source

code for the more popular benchmarks (i.e. Quicksort, Savage etc.) At the same time I tried to become more familiar with the principles of Benchmarking CPU's.

THE RACES

Benchmarks can be described as "Horse Races" of a sort, and like horse races they can be fixed by "doping up" a horse. When I say "doping up" I mean giving the machine an unfair advantage by not performing the test under less than comparable conditions. On the whole, benchmarking CPUs and computer systems is an entire science in itself and the results shown here are not engraved in stone, benchmarks are as much a test of the machines as they are the compiler. Some examples of how a compiler can affect a result are, how efficient is the assembly code that is generated, does the source code fully utilize the math libraries that are available on certain chips, are co-processors involved, contentions between 16-bit and 32-bit buses, and so on. The source code for the benchmarks gave some indication that no apparent unfair advantages were taken. For example, variables could have been declared as register variables on machines with a large register set invariably speeding up result times. This is not the case here, but it could be done. What I'm trying to say is that before anyone quotes performance specs to you find out what the conditions of the test were. Even the worlds worst machine can come off looking like "The Roadrunner".

Benchmarks that were performed on the A1000 Amiga's were compiled using the Manx Aztec C V3.4a compiler. All benchmarks, source and executable codes, except the Dhrystone are available on this month's AmigaTimes disk.

Now a little explanation about the benchmarks. Fibonacci numbers are a sequence of numbers each of which, after the second, is the sum of the two preceding numbers: i.e. 1,1,2,3,5,8,13,21,.... The process is repeated 100 times in the Fibonacci Benchmark. The Float benchmark performs 7 double-precision multiplica-

tions and 7 double-precision divisions, this process is then repeated 10,000 times. The Sieve of Erastosthenes Algorithm is employed in the Sieve Benchmark for computing 1899 prime numbers. The Sort benchmark uses the Quicksort Algorithm 100 times on an array of 1000 long integers. The Savage benchmark is a floating point test using a nested sequence of trigonometric and transcendental functions in a loop of 25,000 iterations.

THE RESULTS

Test	Num. of iterations	Amiga A1000	Amiga A1000 w/68010	Mac SE	IBM-PC AT (8MHZ) without FPU
Fibonacci	100	251.85	246.92	264.00*	950.00
Float	10,000	19.44	17.92	229.98	116.36
Sieve	100	63.06	62.24	64.70*	26.71
Sort	100	93.06	88.42	111.30*	46.53
Savage	25,000	102.02	108.20	1884.30**	1103.00
* 10 iterations were run and results were scaled accordingly					
** 2500 iterations were run and results were scaled accordingly					

Now that it is known that "horse races" can be fixed, how are we to interpret the results obtained? Well, let's just say that the A1000 Amiga (with and without a 68010) was able to run some of the benchmarks faster than the Mac SE, the only machine that is close to its price range. Under similar conditions the Amiga 1000 finished with respectable results amongst the other higher priced machines.

THE VERDICT

The conclusions that I've reached from this experiment in amateur hacking is that I would not recommend that you go out and immediately purchase an 68010L8, unless you are in the position to take full advantage of its capabilities, something that advanced programmers may be able to do. You see it was discovered that the performance of certain programs such as the aforementioned DrawPlus, were enhanced by the 68010 while others would not run at all. As mentioned earlier, one of the potential software based problems surfaced. One

reason behind the inability of certain programs to run is that some of these programs use the instruction "MOVE to SR" (Move to Status Register), a assembly code instruction that is valid in user-state with the 68000, but invalid in the user-state with the 68010. Execution of "MOVE to SR" by a program running in the user mode on the MC68010 will result in a privilege violation causing the system to crash and invoking the dreaded Guru. The "techies" know what

this means but to a layman such as my humble self all this means is IT DON'T WORK.

A remedy for the "MOVE to SR" problem would be an exception handler in the or "patch" program. The program called DeciGEL was discovered on a public domain disk, this program effectively functions as an error trapping "wedge" by that intercepts the instruction privilege violations. DeciGEL examines the instructions, and when it encounters the "MOVE to SR" instruction it would simulate the instruction as appropriate. The instruction is replaced with the "MOVE to CCR" (move to condition codes) which is a valid instruction in the user state with the 68010.

Now the verdict; the marginal increase in system speed for CAD applications or math intensive programs could sort of justify the change, but there is no real need for the transplant. For those of you who are still intent on venturing ahead and giving the 68010L8 a try, keep in mind that there

is the occasional odd program that will not run even after using DeciGEL, one such exception is Barbarian TM from Psygnosis Ltd. Barbarian simply would not work leading one to conclude that the problem does not only lie with the aforementioned illegal instruction but could be any one of those listed below. Programmers that are still entertaining the idea of a 68010 exchange, the current versions of the Manx Aztec C and the Lattice C Compilers do not generate the "MOVE to SR" instruction. Commodore, in the persona of Carolyn Schepner has given a few guidelines to facilitate software compatibility between the M68000 family of CPU's, for those that have not looked at the September/October issue of the Amiga Developers Newsletter, AMIGA MAIL, the guidelines are as follows:

1. Do not use the upper 8 bits of a pointer for storing unrelated information. The 68020 uses all 32 bits for addressing.
2. Do not use signed variables or signed math for addresses.
3. Do not use InitStruct() to initialize APTR fields. For 68020 compatibility, InitStruct() does not use 24 bit APTRS. Use ULONGs instead.
4. Do not use the MOVE SR,<ea> instruction. This is a privileged instruction on the 68010/68020. Use the Exec function GetCC().
5. Do not execute code on your stack.
6. The assembler TAS instruction should not be used in any Amiga program. This instruction's indivisible read-modify-write cycle can conflict with system DMA.

THE WARNING!!!

DISCLAIMER DISCLAIMER
DISCLAIMER DISCLAIMER
DISCLAIMER DISCLAIMER

This hardware hack will absolutely, positively, and without a doubt void your Commodore Amiga Warranty, the Author and Editing Staff relinquish all responsibility for any damages or injuries that may be incurred from information obtained from this article. ☐

AMIGOTIMES IMAGE CONTEST

AmigoTimes is celebrating its PREMIERE issue with the announcement of an Amiga Image contest. Send us your graphic interpretation of the topic below.

TOPIC: "Space Travel"

PRIZES:

One (1) First prize: Telecommunication System.

Avatex 1200E Modem

Connecting cable

Terminal emulation program: **Dr. Term Professional** by *Progressive Peripherals & Software, Inc.*

Total retail value: US \$214.95

Three (3) Second prizes: Ferrari Formula I by *Electronic Arts*

Total retail value: US \$49.95(each) Software package

Five (5) Third prizes: A six month subscription to AmigoTimes

Total retail value: US \$41.70(each)

AMIGOTIMES "AMIGA IMAGE CONTEST" RULES:

1. To enter contest, complete an Official Entry Form and mail it to:
AMIGOTIMES
Amiga Image Contest, P.O. Box 1228
Ville St.Catherine, Quebec, J0L 1E0
CANADA
2. Photocopied or hand drawn entry forms are eligible.
3. Enter as many images as you wish.
4. All submitted art work must be original.
5. The use of digitizers is permitted, but no digitized photos please.
6. Pictures can be in any IFF format or HAM mode.
7. All images must be presented on a 3.5" disk.
8. Images must include an explanation of how the image was created along with a list of products used.
9. Entries received must be post dated no later than midnight July 15th, 1988.
10. Entries are judged by the **AMIGOTIMES** Staff whose decisions are final.
11. Contest results will be published in the August AmigoTimes issue. Winners will be notified by mail.
12. All prizes will be awarded on August 15th, 1988.
13. Contest is not open to AmigoTimes staff members or their families.
14. Any inquiries regarding the legitimacy of the contest, and or the awarding of prizes should be directed to the:
Regie des Loteries et Courses du Quebec
15. For a list of prize winners, or additional copies of contest rules, send a self-addressed, stamped envelope to:
AMIGOTIMES, Image Contest Winner List, P.O. Box 1228,
Ville St.Catherine, Quebec, J0L 1E0 CANADA

Prizes were graciously donated by: *Avatex Modems*
Electronic Arts
Progressive Peripherals & Software, Inc.

AMIGOTIMES IMAGE CONTEST ENTRY FORM

Name

Address

City

State/Prov.

Postal Code/ZIP

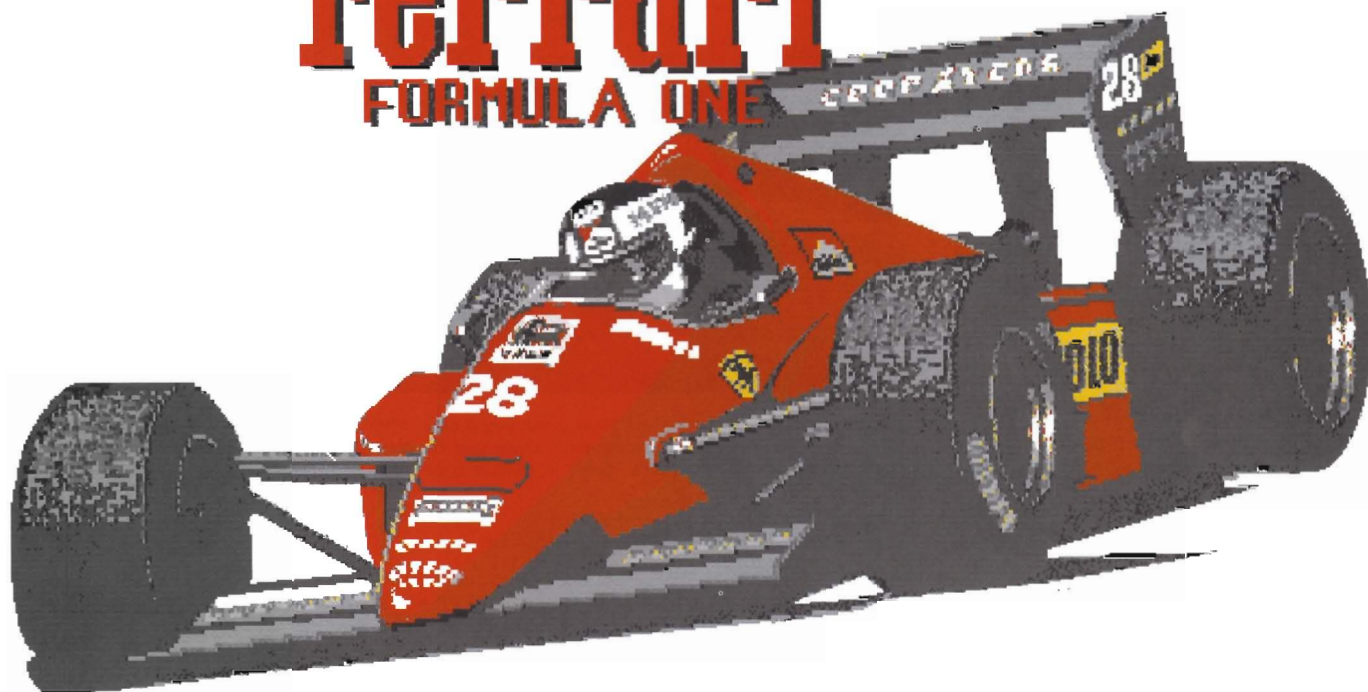
Telephone (Home)

(Business)

All submitted entries become the property of AmigoTimes. AmigoTimes reserves the right to use and/or redistribute all entries as public domain.

Ferrari®

FORMULA ONE



Attention sports fans! The 1986 Formula One season is now open courtesy of Electronic Arts. If you are a die-hard Formula One racing fan, this is the game for you. Ferrari Formula One soundly integrates just about every aspect of Formula One racing, from the grueling real-time races to the injuries incurred from a major crash that could leave you "laid up" for the season. You have a choice of being an active participant i.e. driver of a car, or a passenger that can still make race altering decisions. As an active participant in what is referred to as a Formula One simulator, you make all the decisions, selecting tire compounds, wind tunnel testing, engine testing, turbo-boost adjustment, etc. From the start you are lured by the packaging for Ferrari Formula One, the cover has the most breath-taking creature that you will ever lay your eyes on, a blood red Ferrari in all its splendor. That's right

sex sells, no second-rate packaging here. The package consists of the Program Disk, the Drivers Reference Guide (a condensed manual and keyboard guide), and the Ferrari Formula One manual.

THE MANUAL

Reading through the manual alone is an education on the sport of Formula One racing, it is both concise and well-written. The manual clearly states at the beginning that if you do not wish to use the advanced features of Ferrari Formula One, but want to start racing right away, all you have to read is a small section entitled "GOING FOR A LAP" and you're off and running. But you will quickly learn that in order to truly re-live the Formula One experience you must read the manual. The manual is designed in such a way as to allow you to use as much of it as you want. It is quite instructional in explaining the

By Olusegun Olaniyan

sometimes cryptic numbers obtained during the time trials and it also does a good job at explaining the meaning of a lot of icons and symbols that appear on the screen. The manual gives tips on driving, such as negotiating those nasty hairpin turns with the latest techniques sanctioned by the pros.

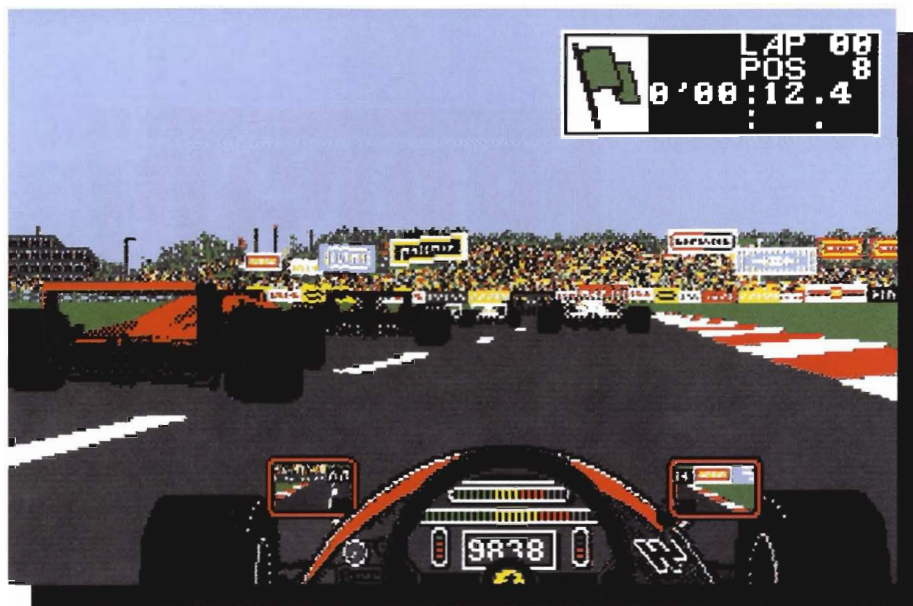
THE GAME

The distinctive feature of this game that I feel will make it a big seller is its recreation of the 1986 Formula One Season. It recreates it in the sense that the top drivers of that 1986 season will finish exactly as the season ended, but if you are up to the challenge, you actually have a chance to upset those season results and in fact you could become the Formula One champion of 1986. Don't get it into your head that the history books will be re-written, but you can actually say that you gave Alain Prost, Keke Rosberg, Nigel Mansell et al. a run for their money.

The player interface for this game is very good, everything is virtually "point and click" with the mouse, meanwhile the cursor changes to the icon that represents the area of the game that you are in. For example, at the Fiorano (pronounced FEE-OH-RON-OH) Test Track, your home base, all that's needed to go to the various areas of the test facility is to aim the cursor onto a building to enter it, or the track to take the car out for a test run. To pack up and go on the road just point at the truck and your're ready to race at other world-famous tracks.

GETTING STARTED

The race control requester is where you select the race settings such as track, session, race length, skill level, and driver name. When you choose the track control, a list of the tracks for the 1986 season are displayed. The skill level control allows the player to set the level of difficulty, from Formula Three (F3 - the easiest) to Formula One (F1 - the hardest). The whole idea is that the higher the skill level, the more skillful your opponents become, and the greater the chances that wear and tear will cause a mechanical failure in your car. At the Formula One level all shifting



The cockpit view of the Ferrari racer gives you a clear view ahead, also, notice the view through the twin rear-view mirrors. The instruments show you remaining fuel, turbo boost pressure, RPM, water temperature, and oil pressure. There is even a gear stick which shows the present gear position.

and turbo boost adjustments are controlled manually.

Driver control allows you to enter your own name as a driver for the season, this can be saved along with your season results on a blank initialized disk. After going through this formality you are now ready to begin your racing career.

PRE-RACE

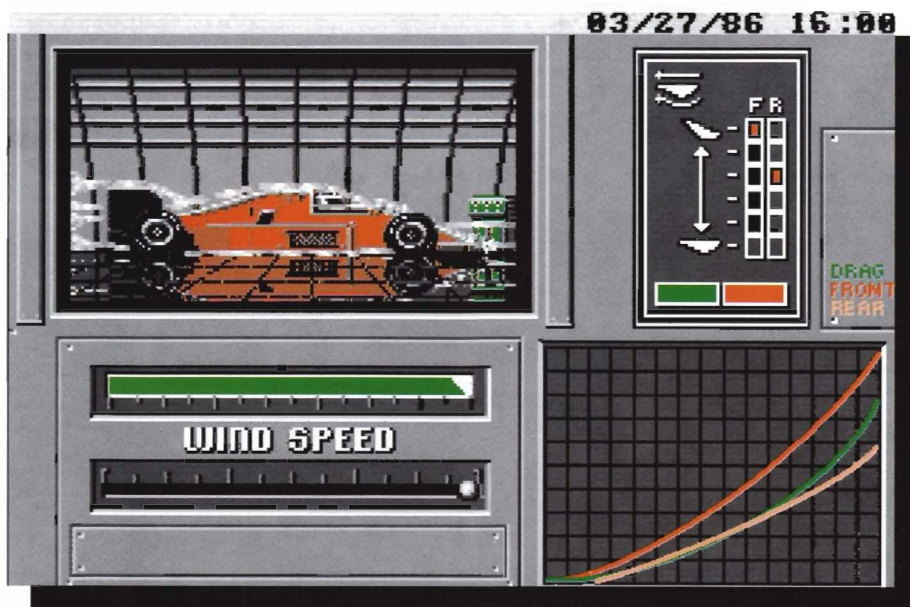
The garage is where all major adjustments are made to your Ferrari before proceeding to the test track or to a race. In order to get your car into full driving trim it will be necessary to make adjustments to suspension, gear box, tires, and engine. At this point unless you are very familiar with the intricacies of "wind tunnel" testing and using a "dyno room", it is highly recommended that you read the manual. The manual gives an explanation of what adjustments to make when doing wind tunnel testing, or how to properly break in an engine in the dyno room. The manual also gives hints and tips on suspension, gearbox selection, turbo chargers, and an enlightening tutorial on tires and their compounds.

Now if after reading the manual, you're still having a hard time telling a racing slick from Grace Slick, you will be happy to know that you have a per-

sonal mechanic available to give you advice on suspension, transmission, tires, whatever might concern your car. Your mechanic's name is Mauro, a silent computerized crew chief. To get Mauro's suggestions on car adjustments, when your car is in the pit or garage simply select the toolbox, all the control panels for the car are displayed with Mauro's suggestions all highlighted in bright green.

Upon entering the test track you can familiarize yourself with the controls of the "unleashed motorized beast" that you are now sitting in. The controls are fairly simple with the left mouse button being the brakes, and the right mouse button being the accelerator. Shifting is done for you automatically in the demo stage but as the levels become more advanced the shifting as stated before is manually operated. Continuously shifting and adjusting the turbo boost takes a fair amount of co-ordination, needless to say it helps to have the reflexes of a Mongoose.

To say that the steering is very responsive is an understatement, in fact I don't know one member of the staff that didn't trash their car the first time out. Steering takes some getting used to, but after some time and a couple of hundred grand worth of ruined racing machinery we all got the hang of it. As a word of



The Wind Tunnel room is where you set your Ferrari's aerofoils. Both the front and back spoilers can be adjusted thereby allowing you to get the most effective drag coefficient for a particular track. What may be a perfect setting for one track may be totally wrong for another so you'll be going through a lot of testing by trial and error. The wind speed inside the tunnel can be adjusted by a slider.

warning though, go slowly and take curves at a sober speed because these cars are easy to pile up.

After you have familiarized yourself with the controls and are confident enough in your abilities as a driver, it is now time to go on the road and test your abilities against the pros. Your first race of the season is the Brazilian Grand Prix in exotic Rio de Janeiro. Time is also a critical factor because when you travel you must also take into consideration travel time. For example the first race takes place on the 23rd of March, therefore to get from Fiorano to Rio in time for the race you should leave Fiorano no later than the 19th of March. Keep in mind that you can travel to competition tracks before or after the actual races, although you will only be able to test the track.

THE RACE

You are now at your first Grand Prix race, at this point you will find that the race is broken down into various track sessions. There are a total of six track sessions spread out over three days, two sessions per day. Day one involves going through the first practice session where you familiarize yourself with the track, you will then be trying to qualify for a good position on the starting grid.

Day two is a repeat of day one, you get another practice session and are now try to improve on your qualifying time. Keep in mind that on day two things really begin to get hairy because all drivers are not only trying to beat their own qualifying times, but they are also trying to beat everyone else's time. Day three is race day, the day to separate the men from the boys (or the women from the girls). The day begins with a 30-minute warm-up session in the morning this is followed by the race in the afternoon. The race is nothing short of spectacular, all the top name drivers are here and they aren't here to play shuffle board. The second the green flag is up everybody is off the starting line "like a bat out of hell", you're continuously up-shifting and down-shifting for your honor and for your life. During the race you must know when to play around with the Turbo boost gauge for more power, but keep in mind you have a limited amount of fuel to finish the race, there's nothing more disgraceful than running out of fuel before the race is over. Always keep one eye on the rear view mirror, who's that coming up from the rear, could it be Alain Prost trying to challenge you for the lead; obviously the fumes from his fuel have made him brave, this is the time for some very

strategic maneuvering to lose him. In the end you emerge victorious and ask yourself why do you race, for the money? No, you race for the glory, the fame, the thrill of mano a mano combat, and of course, La Dolce Vitae!

Winning a race is decked out with all the regalia of the real thing complete with trophies and giant bottles of bubbly. I can't honestly say that I have been able to win the Grand Prix for the 1986 Season, but that's not from lack of trying. The beauty of this game is that you are free to race as often as you wish, in any race and on any track.

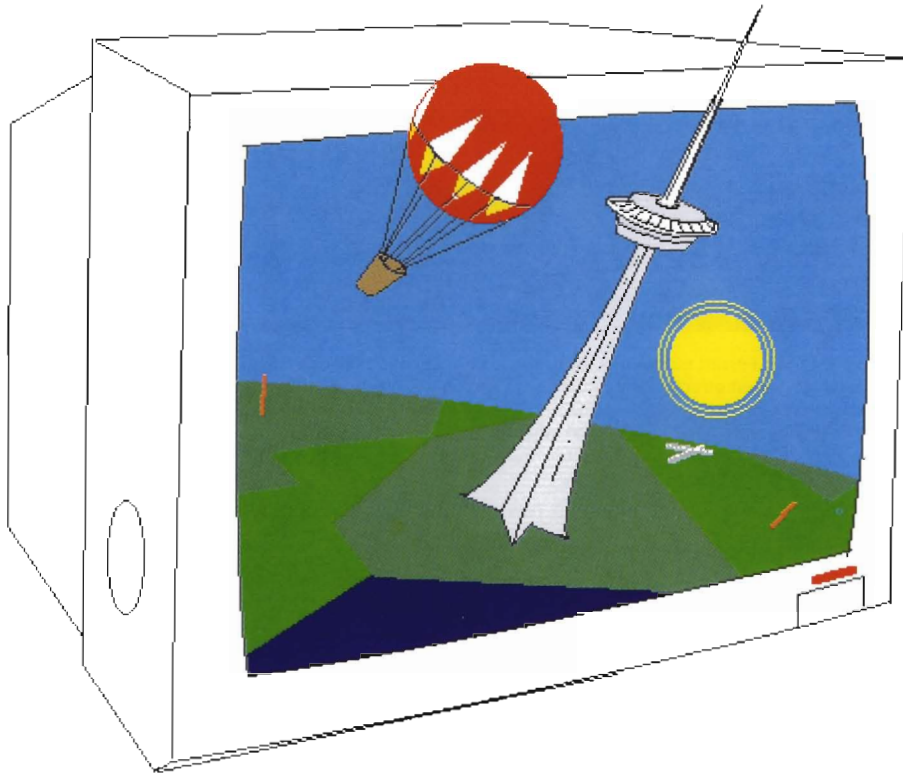
CONCLUSION

After playing with this Formula One "simulator", the only things missing will be your fake accent, an arrogant air about you, and your leather driving gloves. You might also find that after a while you begin to take Ferrari Formula One a little too seriously and the people around you will begin to say things like, "It's just a game", but then that's like saying a Ferrari is just a car. □

Ferrari Formula One Electronic Arts

1820 Gateway Drive
San Mateo, CA 94404
(415) 571-7171
\$49.95
copy-protected
Requires 512K

SubLOGIC SCENERY DISK #7 AND #11



Last year, SubLOGIC released the first of a series of Flight Simulator II Scenery Disks. It contains a larger area of scenery, enhanced graphics and colors superior to the standard Flight Simulator II (FS II) scenery.

Two disks are currently available for the Amiga. The first, No.7, covers almost the entire eastern U.S. coast from Atlantic City straight down to the Florida Keys, and part of the Bahamas. It's followed this year by No.11, which expands the SubLOGIC "world" even more with the eastern half of the Great Lakes region.

SCENERY DISK NO.7

There's no disputing the fact that the area covered by this Scenery Disk is huge. Almost the entire U.S. eastern

seaboard is yours to explore. In fact, the area is so large that if you were to use the Cessna to fly from the northernmost city down to the Bahamas, it would take almost half a day to complete the trip. If shorter and less-involved flights are more to your liking, there's an abundance of airports (around 140), cities, and islands. Some major cities that have special scenery are Miami, Tampa Bay and Washington. And if you've ever wanted to go to Andros Island you'll find it just past the Straits of Florida in the Bahamas.

THE VIEW

When you compare the overall graphics and geographic detail of a Scenery Disk with the FS II, it's obvious that both are similar except the Scenery Disk is definitely superior. The improvements are

By Michael Hermann



Buzzing the Shuttle launch pad at Cape Canaveral. Scenery Disk #7 has no restricted air space, you can even land on the front lawn of the White House.

immediately noticeable once you're up in the air. The standard FS II terrain color of one shade of green is replaced with light and dark green patches that add a greater sense of realism to the environment. Metropolitan areas are now distinguished by grey areas. These improvements make flying by VFR (visual flight rules) more realistic than ever before.

Sight-seeing with the simulator is one of its main attractions and this was not forgotten by the designers of this disk. You'll get a real kick out of two places in particular: Washington, D.C. and Cape Canaveral in Florida. In real life, it's not advisable to fly over either of these areas but with the simulator it's no problem. There are no military defence systems around Washington so you can easily buzz the White House with your learjet going over 500 knots. Or, if you prefer a more civilized approach, you'll really enjoy touring the nation's capitol with the Cessna. Remember to stay below 2500 feet because the buildings aren't visible past that altitude.

Cape Canaveral (located east of Orlando International) is another restricted area where you can fly through uninhibited by FAA restrictions. Standing ready on the launch pad is a space shuttle along with an external tank and two booster rockets fastened to its underside. Just southeast of the launch area is a row of rocket gantry towers which is a lot of fun to use as an aerial slalom run. If you're a good enough pilot, switch to the learjet and try to land (at around 200 kts) on the shuttle's extra-long (5.5 km) runway.

DOCUMENTATION

The documentation contains the few instructions needed to use the disk. In addition, you'll find four sectional directories (that contain runway and radio information for each airport), suggested flight routes, nine ILS (instrument landing system) approach procedure charts and four sectional maps. The sectional directories are easy to use as reference guides and can be stored in a handy three-ring binder (available from Sub-LOGIC) that can store the disks, directories, instructions, and maps for up to six Scenery Disks.

SCENERY DISK NO.11

Almost everything was improved for this disk and a couple of new features were thrown in as well. While it's not as large and doesn't have as many airports (about 84) as No.7, there's a very large area to fly around in. It covers the eastern half of the Great Lakes region including Lake Erie, Lake Huron, Lake Ontario, and Georgian Bay. There are also a lot of cities in the U.S. section (Cleveland, Detroit, Pittsburg, etc.) and

also some cities across the border in Canada such as Niagara and Toronto. In the southeastern section of Ontario there are lakes and rivers practically everywhere.

There's a terrific new feature in the Niagara Falls region. To get to the falls by way of the river, take-off from St. Catherines airport and head east along the shore of Lake Ontario, then head south over the Niagara River towards the city (look for the Skylon Tower). Stay about 200 feet above the water and you'll soon enter a river canyon whose walls (about 300 yards apart) will flank you on both sides. A certain degree of skill is required to avoid clipping the plane's wings and falling into the drink. For most FS II pilots, it shouldn't be too difficult to follow the river upstream as it meanders its way towards the falls. When you finally reach the falls you'll have a few seconds to play tourist before you have to fly out of the river gorge in order to avoid crashing into them.

When you're flying around Buffalo and you see another plane in the sky, don't worry, you're not seeing things. Well, you are and you aren't. The plane (actually a sailplane) is there, but it doesn't move. If you pass by it from a certain direction it will give the illusion of movement. I found it interesting to have another plane (there are three more at other locations) in the sky but it seemed a little silly that the plane is frozen in mid-air. There is also a hot-air balloon around Grand Lake, Ohio. The balloon is also suspended in the air but I guess balloons don't always move very fast anyway.

Another new feature is the "trans-



On Scenery disk #11 you can fly all the way up the Niagara river canyon, after which you have to pull up hard to avoid the falls.

"The Ultimate Computer Baseball Simulation"

★★★★ + Four Stars Plus

Info Magazine

EARL WEAVER

Baseball™

By Eddie Dombrower,
Teri Mason and
Earl Weaver

The Critics Report...

"Game of the Year"
Computer Gaming World

★★★★ Graphics
★★★★ Gameplay
"The definitive baseball game available today for any home computer."
Computer Entertainer

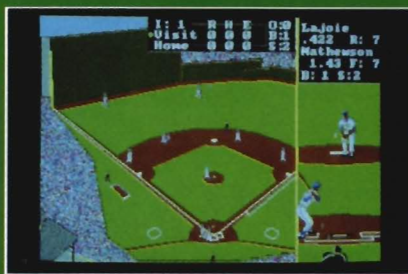
"It's all there. You can trade, set up leagues, even ask Earl for advice."
Computer Shoptalk

"Everything a dedicated baseball fan and computer gamer could want."
Commodore Magazine

"It's baseball the way you want to play it."
AmigaWorld



The Green Monster at Fenway. At 315 feet it's a hitter's dream. One of thirty-two historical, actual, and fictitious ballparks.



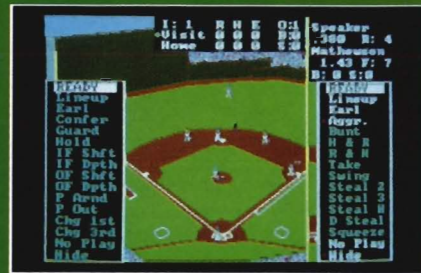
Hit from the batter's viewpoint. Read pitches by the seams on the ball. Real physics affect the spin, trajectory, and play of the ball.



ELECTRONIC ARTS®

Real Baseball

- ★ Match up the best from any era. Sandy Koufax in '65 fires blistering heat while Gaylord Perry in '72 junkballs his way to a 1.92 ERA.
- ★ Take leadoffs, hit & run, bunt, steal bases with head first slides.
- ★ Fool runners with pickoffs.
- ★ Pitch around Murderer's Row.
- ★ Guard the lines, trap runners in rundowns. Pitchout on Maury Wills and nail him.
- ★ Shift the outfield. Charge the infield on bunts.
- ★ "Ask Earl" feature gives you advice from modern baseball's winningest manager.
- ★ TV special effects like instant replay & slo-mo.
- ★ Four different playing modes for any level of baseball expertise. From Sandlot to Major Leagues.
- ★ IBM/Tandy version is NOT COPY PROTECTED.



Settle down the pitcher. Call a conference on the mound. Shift your fielders; even ask Earl what he'd do.

Leagues Forming!

"The closest we'll ever get to a major league dugout"
Robo City News

You're the GM

- ★ Trade, draft, even clone players.
- ★ Cumulative Stats Compiler & Player Stats Editor included.
- ★ Print out line-ups and box scores in detail.
- ★ Over 50 hitter/fielder ratings.
- ★ Over 30 pitcher ratings.
- ★ Play on turf or natural grass.
- ★ Over two dozen professional ballparks accurately recreated.
- ★ Build your own dream stadium with the Ballpark Editor.

You're the Coach

- ★ Set pitching rotation through an entire season and playoffs.
- ★ Three manage modes. You do it all, or let Earl call the shots.
- ★ Warm up pitchers. Don't wait until your starter loses it.
- ★ Play Major Leagues, Minor Leagues, Semi-Pro or Sandlot.
- ★ Check the Radar Gun. Ask the catcher how the pitcher's doing.



Official Licensee

Season Data Disks Available

How to Order: Visit your retailer or call 800 245-4525 for direct Visa or Mastercard orders (in CA call 800 562-1112, 8-5 Pacific Time). The direct price is \$39.95 for IBM & Tandy, \$49.95 for Amiga. Send U.S. check or money order to Electronic Arts Direct Sales, P.O. Box 7530, San Mateo, CA 94403. Add \$3 shipping and handling. Calif. add 6.5% tax. Allow 3 weeks for delivery. There is a 14 day money back guarantee on direct orders. Amiga and IBM are registered trademarks of Commodore Electronics, Ltd., and International Business Machines, Inc., respectively.

porter". After logging onto No.11 from the FS II disk, you'll be put in a blank area where you can either use "POSITION SET" in the "NAV" menu to enter the coordinates for the place you want to begin, or, you can select the "MAP DISPLAY" which will "transport" you to the default Detroit Metro/Wayne County airport. Unfortunately, it's a fairly unnecessary feature unless you begin most of your trips from that airport.

The "MAP DISPLAY" and the "out-of-bounds" scenery on No.11 have been improved considerably from the ones on No.7 and FS II. Should you happen to fly out of the boundaries, select the

Toronto, and there are a large number of lakes throughout southeastern Ontario. There's even a runway as far north as Val D'Or up in Quebec.

DOCUMENTATION

Documentation format is virtually identical to No.7. It includes disk-logging instructions, flight suggestions, two sectional maps, two ILS approach procedure charts and two sectional directories.

CONCLUSION

The most enjoyable and interesting thing about using FS II is its ability to simulate, quite realistically in most ar-

FLIGHT SIMULATOR II UPGRADE TO V1.1

by Michael Hermann

On Scenery Disk 11 is the latest upgrade for Flight Simulator II. Follow the accompanying instructions to update your 1.0 or 1.01 version to the 1.1 version. This latest version brings no truly spectacular changes - overall, mistakes were corrected and disk operations were simplified. The cloud error (red and green clouds) found in the earlier versions of FS II, has been corrected to white and grey. A new feature in the multi-player mode allows you to "lock" onto the other player's plane in case you lose sight of each other. Scenery Disk logging has been simplified. To log onto a Scenery Disk from the FS II disk, press "E" and a window will appear requesting you to insert any Scenery Disk in any drive. Do the exact same thing to switch from one Scenery Disk to another. If you are in mid-flight, pause the game before you press "E". To log back onto the FS II disk, press "W" and the program will request the FS II disk. It's that simple.

"MAP DISPLAY" and zoom out to see your exact location on the North American continent. The display shows the entire continent from the top of Canada down to the Panama Canal. Obviously, this makes finding your way back into the "main area" much easier. You can also fly or slew around the entire continent where you'll find "generic scenery", main highways and large rivers (as blue lines only) like the Mississippi.

Some Detroit area sights worth mentioning are Tiger Stadium, the famous "Ren Cen" (Renaissance Center) and the Ambassador Bridge which lights up at night. In Pittsburg, there's a number of buildings in the downtown area. Unfortunately, our copy also had error around there. Lines protrude from one of the buildings and connect to the plane itself. It almost looks like a laser fight, but it's a definite glitch. North of the border you'll find the CN Tower in

eas, the experience of flying an airplane - you can create flight plans, fly by VFR or IFR, fly in bad weather, navigate, etc. In addition to this, you can fly simply for the sake of sight-seeing. Both disks are perfect for whatever aspect you enjoy most about using FS II.

While a Scenery Disk does have slightly more geographic detail than the scenery on the main FS II disk, it's really just an extension of FS II. Adding a Scenery Disk increases the simulator's "airspace" and can only make using Flight Simulator II an even more realistic and fun experience. □

Scenery Disks No. 7 & 11

Sublogic Corporation

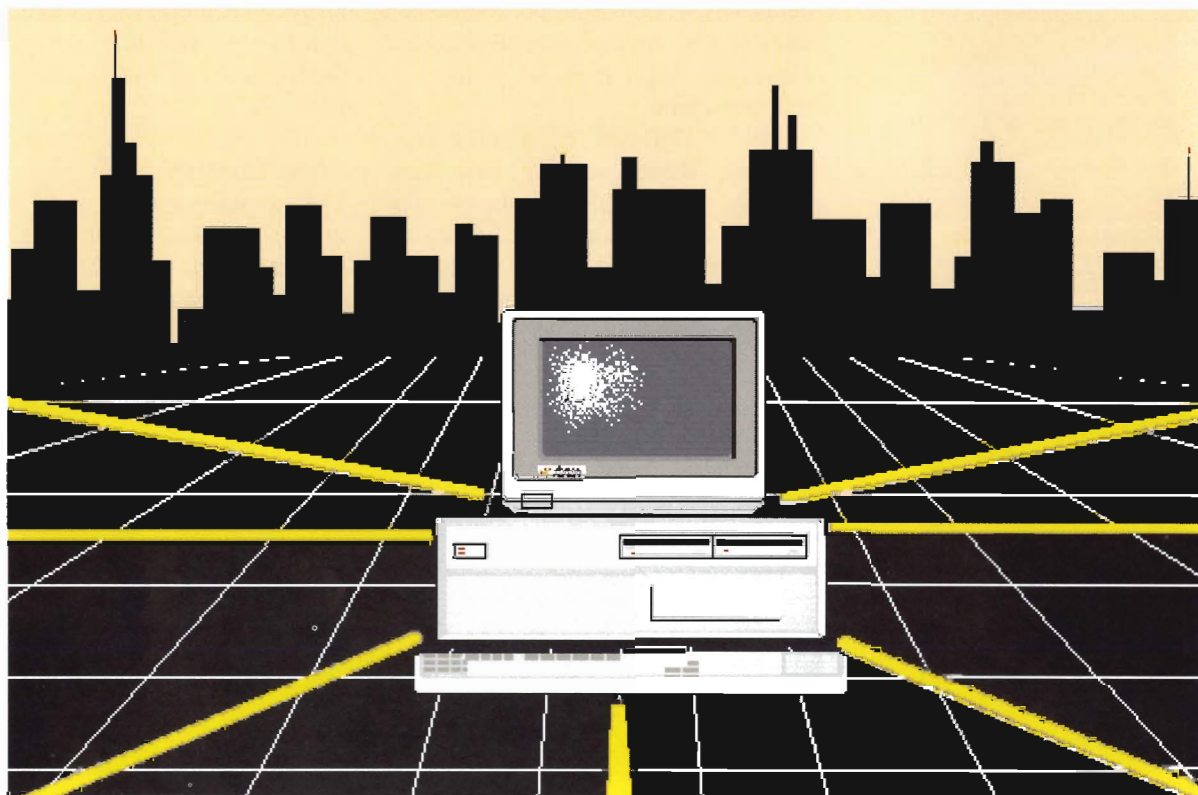
713 Edgebrook Drive
Champaign IL 61820
(217)359-8482

\$24.95 (US)

Not copy protected

512K Required

TELECOMM



An entire universe of Amiga goodies lives on the other end of your telephone line. In this and future columns, I'll explain why you need a modem connected to your Amiga - how it will broaden your horizons, help you become more productive and make the time spent with your Amiga even more enjoyable.

We'll start out with the simple stuff and work our way towards more esoteric and advanced techniques and tricks that will help turn you into a "power user."

For the purposes of this article, I am going to assume that you have not yet purchased a modem, or that you already have a modem but do not know what to do next.

In this article, we'll discuss shopping for a modem, connecting it to your

Amiga, getting the right terminal software to suit your needs (and not melt your credit card) and, finally, going on-line. In future articles, we'll cover topics such as:

- o The kinds of software and other helpful items you can expect to find on local bulletin boards and commercial services.
- o The intricacies of downloading files to your amiga.
- o The mysteries of file compression techniques and how to uncompress" or "unpack" the files that you download if they have been encoded in some way. (Most services do encode their files - an encoded file takes less space for them, and saves you money)

By Harv Laser

(with thanks to Rob Peck)

But let's begin at the beginning. What is a modem, and why do you need one? The word modem comes from "MODulate/DEMODulate." Put simply, a modem is a hardware device that allows computers to communicate with each other. Think of it as your computer's mouth and ears. You connect a modem to your Amiga with a cable, connect a telephone line to your modem, and you're in business.

With a modem you can not only talk to other computers by typing on your keyboard, but you can listen to other computers and receive information from them. The modem does all the dirty work of translating computer data into audio tones that can be sent down an ordinary telephone line to another computer whose modem will listen to those tones and translate it back into data which it can understand.

Modems can operate at different speeds, known as "baud rates." During the early days of telecommunications for personal computer owners, 300 baud was the most commonly used speed. To give you an idea of how fast 300 baud is, a person who can type very quickly, about 100 words per minute, is typing at nearly 300 baud. The higher the baud rate, the higher the rate at which data can be transmitted across your phone lines.

Early modems were large, expensive and clumsy. Like most of the late 20th century's electronics modems have become smaller, less expensive, and much easier to use. Today, the most common baud rate for a home computer (and for many commercial services and local bulletin boards) is 1200 baud, or four times faster than 300 baud. We are now in another period of transition when many computer hobbyists are beginning to use modems capable of communicating at 2400 baud. There are even modems that can run at up to 9600 baud.

When shopping for a modem, don't just consider the price of the modem itself - also consider what your own personal time is worth. Note that the cost of a 2400 baud modem is only marginally higher than a 1200 baud modem from the same manufacturer. Since the 2400 baud modem can trans-

fer data twice as fast as the 1200 baud modem, you may be able to get the same work done in only half the time. 2400 baud modems have reached well into the "Under \$200" price range now. 9600 baud modems are still in the \$500-\$1000 and up range.

When you buy a 2400 baud modem, you can also operate at 1200 and 300 baud. (Many privately operated computer bulletin boards don't yet have 2400 baud capability, for example).

When shopping for a modem, it's also important to make sure that the unit you're considering adheres to the "Hayes AT Command" standard. Hayes is a manufacturer of computer modems who years ago established a set of protocols known as the "AT" commands. Other modem manufacturers followed suit and emulated those commands in their own modems, so the Hayes standard became to modems what the Epson standard is to printers. Most telecommunication software is written to take advantage of the Hayes "AT" commands just as most software that sends output to a printer is written to make use of the Epson printer escape codes.

Hayes still makes and sells very capable modems. You'll find, however, that they tend to be rather expensive compared to other brands. Two extremely popular brands of modems currently in favor with many Amiga owners right now are Avatex and Supra. Both of these companies make small, inexpensive, full-featured modems which perfectly emulate the Hayes AT command set.

Since Avatex began manufacturing modems, they have created several models. Their first 1200 baud model, known simply as the Avatex 1200, did not completely conform to the Hayes standard and caused some difficulties with certain modem software that expected the modem to respond as though it was a Hayes modem. Because of those incompatibilities, I would instead recommend the Avatex 1200hc (the hc stands for "Hayes compatible"), or their newer models, the Avatex 1200e and the Avatex 2400. Both of these are also fully Hayes compatible and include some of the latest "bells and whistles" that make it easier to use a modem. For

example, the Avatex 2400 includes a speaker so that you can hear the modem dialing your call and making a connection to the remote computer. It also includes 8 front panel status lights to let you know in which mode it is currently operating, and a long list of other standard features at a reasonable price.

Supra has recently introduced their new 2400 baud modem with features similar to the Avatex 2400, but in an even smaller, lighter case. The Supra modem also contains far fewer chips and thus produces virtually no heat when it is operating.

Both modems have features such as "non-volatile memory" which can store modem configurations even when powered off, and software control of many settings, such as the volume of the modem's speaker. Older modems force you to set tiny "dipswitches", just like most printers have, which can be a hassle - these new generation modems are much easier to set up and operate since all of the functions formerly set with dipswitches are now controlled with relatively simple keyboard commands.

Due to the tighter requirements for data transfer at 2400 baud, modems that operate at this higher speed have to include better noise filtering than those that operate only at the lower speeds. If you have had some experience with line noise (strange characters on the screen) when communicating, you would notice that these 2400 baud modems offer a very "clean" signal, and you'll see much less line noise creeping into your communications sessions when you use a 2400 baud modem, even when operating at lower speeds than 2400 baud.

If you don't already own a modem, I recommend that you consider purchasing a 2400 baud model. You may not have immediate use for the higher 2400 baud speed but it won't cost you much more to have that capability to begin with, and it won't take you long to realize that the faster speeds will make you more productive.

Amiga files and programs tend to be large. One HAM Interlace picture can take as much as 100K of disk space to store. A program which creates those pictures might be 300K or even larger.

I've seen public domain games even larger than that. When it comes time to start moving these files across phone lines you will want to do so as quickly as possible.

Depending on which model Amiga you own, make sure you get the proper cable to connect your modem to your computer. The Amiga 1000 has a serial port connector different from the 500 and 2000, and it carries electrical voltage on a few of the pins on the serial port. Buying a standard modem cable that has not been designed or modified for use on an Amiga can be extremely hazardous to your equipment's health. You could even do permanent damage to your computer, your new modem, or both, by attaching the modem with a cable with the wrong pin-configuration. Generally, Amiga dealers understand this problem and will sell you a cable which will work correctly with your modem.

A dealer who does not sell Amigas may not be aware of this potential for disaster, so shop carefully and understand that not just "any old cable" will work. You can buy a generic modem cable off the shelf at an electronics supply store, such as Radio Shack, but you will have to modify the cable after purchase to make sure that the pins on the Amiga's serial port which carry voltage do not transmit that power to your modem. This is what I did long ago when I connected a modem to my Amiga 1000. The documentation that comes with your Amiga will have pinout charts which will show you which pins carry voltage from the serial port.

Remember, any time you are connecting or disconnecting cables and peripherals to your Amiga, the power on both the computer and the external device should be OFF. This is extremely important and will help you avoid a lot of heartbreak, not to mention expensive repair charges if you accidentally "fry" one of your components by hooking them together when they are powered on.

If you decide to purchase the Commodore 1680 Amiga modem, you need not worry about any of this, since that particular modem was designed to be powered by those "hot pins" on the

Amiga's serial port and, in fact, comes with a cable permanently attached, ready to plug right into your Amiga. The 1680 modem, however, only operates at 300 or 1200 baud and cannot be modified to work at 2400. The 1680 also has no status lights. So shop around for the best price and make sure you get the correct cable for your computer if you decide to buy an Avatex, Supra, or any modem other than the Commodore 1680.

Once you have your new modem and have correctly connected it to your computer, you'll also need to plug your telephone line into it. Your new modem's instruction manual will explain how this is done. Remember, when your modem is online, connected to a BBS or a network, it's just the same as if you were making a telephone call. You can only use the line for one thing at a time. You can't make a voice call while using your modem, and vice versa.

If, after a while, you find yourself becoming a heavy telecommunications user, consider calling your telephone company and having a second phone line installed, just for your modem. Depending on whether you have a family or not, (other people who want to use the telephone while you're using your modem), a second phone line could restore tranquility to your home life after you've become an online "junkie." The price of installing a second line varies from state to state and is set by your telephone company.

Your next task, after purchasing and setting up your modem, is to get a terminal emulation program, also known as communications software or simply a 'term program.' This software will turn your Amiga into a telecommunications terminal, and is the only way your computer will be able to talk to your modem, to command it to dial numbers for you, and to establish the communications link.

There are many excellent term programs out there, some of which are commercial in nature; you purchase these from a software retailer like any other program. "Online!", "Diga!", "A-Talk", "Dr. Term Professional", and others may be familiar to you if you've

read many Amiga-oriented publications.

Other term programs are available for downloading from BBSs and national networks, or through purchase of public domain disks and from user groups. Some of these programs are public domain meaning they are completely free of any charge. Some are sold on a "shareware" basis - you acquire the program and try it out. If you like it, you send the author of the program his requested donation. "Access", "Amic Term", "Handshake", "VT-100", "Wombat Term" and others fall into this category.

If you need a terminal program, you need look no farther than the disk that was included with this magazine. The disk includes a public domain terminal program that I believe is one of the best terminal programs that you'll find anywhere for the Amiga. It is called The Communicator, version 1.34, also known as COMM.

COMM was written by Dan James (DJJAMES on People/Link) and is an extremely capable piece of software. You will find documentation for COMM on the disk as well, and you should read the documentation file carefully before trying your first online session and print the docs for future use (if you have a printer). COMM is not a complicated program to use, but like any powerful piece of software, it will help you greatly to know what it can do before you start using it. One of COMM's major advantages over other term programs is that it is small in size, and not a memory hog. You will be able to multitask quite nicely with COMM even if you only have 512K of memory in your Amiga.

Depending on how many disk drives your Amiga has, you may want to make a special disk and relabel it (using the AmigaDos relabel command, or the Workbench rename menu item) to be your "COMM" disk. COMM will need to be able to find the "serial.device" software located in the DEVS drawer of your boot disk so that it can talk to your Amiga's serial port and your modem.

You'll also be able to set up an instant phone book of up to 44 numbers you can call with COMM using only your mouse, as well as program up to

DESIGNTEXT™

WYSIWYG

- Full Width 80 column display
- What you see is what you get
- On screen Headers, Footers and footnotes
- Multiple Fonts
- Works in both Medium or High Resolution using Overscan
- On-screen page numbering
- Horizontal Scrolling up to 255 characters

USER ORIENTED

- Gadgets, Menus or Amiga Key operation
- Format Disks and Delete Files from within DesignText
- Run other applications on DesignText screen
- Various Default Settings definable by user
- Definable Function Keys
- Fully Multi-Tasking

PEOPLEBASE

- Fully integrated with DesignText
- Mail Merge
- Address List Printing
- Print Mailing Labels
- "Little Black Book" printing
- Password Protected files
- Multiple search criteria
- Free form comment fields
- Contact history

FORMATTING OPTIONS

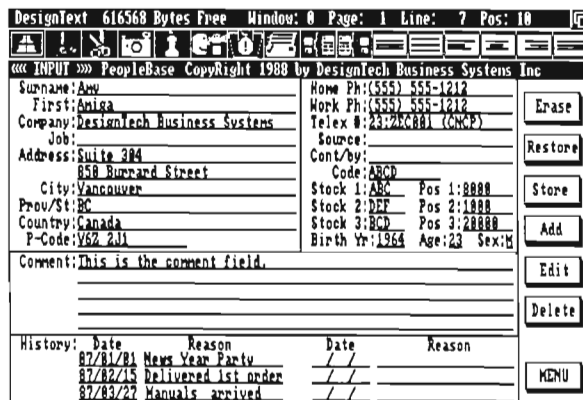
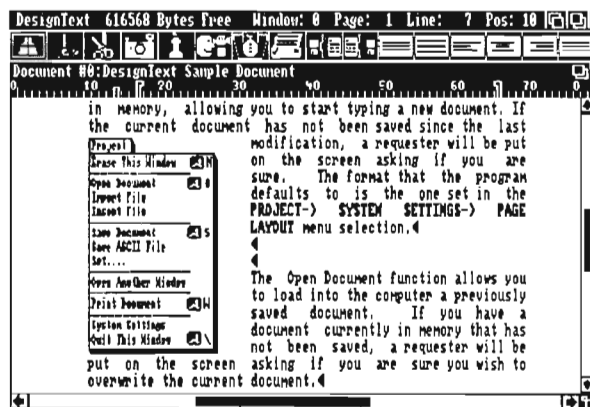
- Multiple Columns on screen
- Hard and Soft page breaks
- Widow/Orphan protection
- Automatic Page Numbering in Arabic or Roman
- Cut and Paste of Tables of Data
- Greek, Math, Cyrillic and Symbol fonts
- Easy to use margin settings
- SET AT CURSOR mode for typewriter users
- Insert or Overstrike modes

SPECIAL FEATURES

- Available in both English and German Versions
- Import IFF Graphics
- Import Textcraft, Scribble and Word Perfect Files
- Extra fast screen update and scrolling
- Special printer drivers for better than normal quality
- Automatic document backup
- Background printing
- On-Line help screens
- Password protection of files
- User Definable Utility Menu

EXTRAS

- Create and Edit Outline
- Table of Contents and Index
- 100,000 word spellchecker with Type and Spell
- Math and Sorting Functions
- User definable Phrase Keys
- User definable Printer codes
- ASCII Load and Save
- Uses System Clipboard Device for easy data transfer from other programs



DESIGNTEXT™

DesignText allows powerful word processing features to be accessed with minimum effort and very little manual referral. "User-Affectionate" features include: lines drawn down the margins of a document when moved, the auto-backup function that waits until you are finished typing before it interrupts you, movable icons that allow you to customize your screen, constant notification to the user of what is happening so you are never left with a "sleeping" computer.

DesignText incorporates specially-designed printer drivers to provide high quality output using dot-matrix printers. It also has

PEOPLEBASE™

the ability to print multi-pass graphics and your printer's near-letter-quality text at the same time and provides near-letter-quality text on printers that don't normally support it.

"PeopleBase" is a mail list manager full integrated with DesignText. While writing a letter you can pop up PeopleBase, retrieve the needed record, and with the click of the mouse the address is inserted into your document. For mail merge applications, full search facilities are available as well as an update feature to keep track of the dates and titles of recent documents sent to each addressee.

\$129⁰⁰
U.S.

Name _____

Address _____

City _____

Province _____ Code _____

Your Printer? _____

Your Computer? _____

Additional Hardware? _____

SEND CHEQUE OR MONEY ORDER TO: DesignTech Business Systems Inc. 304, 850 Burrard Street Vancouver, B.C. Canada V6Z 2J1

20 function keys to provide you with "macros" you can use while COMM is running. COMM will take over the use of your Amiga's 10 Function Keys while running, so these macro settings will not be overruled by any other macros you might have set up using other software during your CLI sessions. Once you quit a session with COMM, the control of your function keys is returned to whatever program was using them before.

A special file called "comm.phone" holds your customized phone book, and another one called "comm.keys" holds your function key definitions and COMM will need to be able to find these files when it loads in order to use them during your telecomm sessions. The simplest way to achieve this is to put everything on one disk, in the same directory, and if you don't boot from that disk, make sure your PATH is set to look for that directory. See your AmigaDos manual for an explanation of the PATH command.

The copy of COMM that came with this magazine will have sample comm.phone and comm.key files with it, so you can analyze them and expand on them to create your own customized working environment. This is all explained in greater detail in COMM's documentation file so you are again advised to print out that file and read it carefully to get the best feel of what COMM is capable of, and how to set up your working disks.

I've tried just about every piece of communications software that is available for the Amiga, and COMM is still the one that I use myself for most of time I spend online. Its screen is clean of any gimmicks and gadgets, and it is very easy to learn to use, yet contains many powerful features and capabilities. As a Sysop on a national network, I need a program that doesn't "get in my way" yet has the options I need to get my job done as quickly as possible. I think you'll enjoy using COMM as much as I do. If you decide to become a subscriber to People/Link and join us as a member of the AmigaZone Club, you'll be able to talk directly with Dan James, COMM's author in case you run into any difficulties in understanding or

using his program.

For now, you need not bother yourself with learning about such esoteric terminology as "parity settings" or "stop bits." When you first run COMM, the most often-used settings for telecommunications are already configured for you, and you shouldn't really need to change any of COMM's defaults. We'll cover those settings, and why you might want to change them in a future issue.

The last thing we'll consider this time is - now that you have your modem connected, and have a good terminal program to use, what next? Where can you call and what happens when you make a connection?

There are, at last count, around 600 known and verified operating Amiga-oriented bulletin boards around the country. A bulletin board is software that runs on a personal computer just like yours, which can receive incoming calls and present you with notice areas in which you can exchange messages with others who share your interests, and a library area from which you can download public domain software to run on your computer. Bulletin boards are usually single-line meaning only one person can call the board at a time, and they are run and maintained by a "Sysop" (SYStem OPERator), usually the person who owns the computer on which the board is running, and usually out of his or her own home. The number of Amiga-oriented boards varies as new ones spring up, and old ones go down, sometimes due to lack of interest or money on the part of their Sysops. Running your own BBS may sound like fun, but it involves a lot of time and work to maintain. Some people tire of it after a while and take down their boards. Other Sysops have kept up their BBSs for many years and have gained a loyal following of callers.

Most every bulletin board I've ever logged onto has, somewhere in its data library, a listing of other bulletin boards. Once you get your hands on one of these lists, and have programmed a few of the numbers into COMM and started making connections, you'll find other lists with more new numbers - a snowball effect.

A good idea the first time you're

calling a new BBS is to use your telephone to call the board first, and verify that it is a working BBS number. If a human voice answers the phone, you might have gotten an invalid number. Occasionally incorrect BBS numbers are distributed and some poor soul will get hundreds of calls, picking up his phone to hear modems screeching at him. If you hear a modem answer (it will sound like a high pitched "screech" sound), then you can pretty much rest assured that the number you got was correct, so you should try redialing with your modem to make a connection.

Remember, when you call a bulletin board, it's just as though you were making a telephone voice call to that phone number. So be aware of the limits of your local dialing area, otherwise you might find yourself with a nasty surprise when your next phone bill arrives! It's easy to get hooked and lose track of time when calling boards outside of your local dialing area. The front of your phone book should contain information regarding what prefixes in your area code are local to you, and how much you'll pay per minute for dialing outside your local area.

Your other alternative to local bulletin boards are the national commercial computer networks. One advantage the national networks have over bulletin boards is that while the network's computers are probably located far away from you, you can normally make a local call through a "packet switching network" to reach the service and not have to pay anything other than the commercial network's own access charges. This is how People/Link works.

People/Link is accessed through either the Telenet or Tymnet packet networks which have thousands of telephone numbers located throughout the USA. (Canadian subscribers may have to use another method to connect to People/Link since Telenet is not available to them and Tymnet has limited area coverage in Canada). When you subscribe to People/Link you are sent an information packet containing a listing of these phone numbers and instructions for dialing in and signing on.

When calling People/Link (we'll call it by its nickname "Plink" from now

on) you pay only for the time you actually spend online. There is no monthly maintenance fee or hidden charge billed to you each month. If you sign up for Plink, and don't use the network at all during a month, you will pay absolutely nothing. Prices vary depending on what baud rate you elect to use, and what time of day you call.

The most popular way to access Plink is during the evening hours, after 6:00 p.m. your local time, and on weekends. This is when the network is most active with other Amiga owners who will sign into our club, The AmigaZone, to read and post notices, upload and download public domain software, or join in live conferences with many other Amiga owners located thousands of miles apart, to discuss their favorite topics. Although prices are always subject to change, and you should verify any prices directly with the network in question, as this is written, if you log into Plink during evenings or weekends at 1200 baud, you will be charged \$4.95 per each connect hour you spend online, with a minimum session of three minutes. If the Telenet or Tymnet number you call is not local to you, then of course you pay for that phone call to reach that number.

Plink also runs special "deals" from time to time and if you call to subscribe you should ask the customer service representative about these arrangements which will help you reduce your online costs using Plink even further.

There are, of course, other competing networks, and I'm not suggesting that you immediately sign up on Plink and ignore all the others. My best advice to you as a novice user is to try them all. If you can afford to, get a "starter kit" from each of the major networks which support Amiga users and try a sample session on each one.

See how easy or difficult each network's menu systems or expert user modes are to navigate and use. See how active each network's Amiga area is. Are there many notices posted so that information is always fresh and current? Look in the network's Amiga program libraries. See how new their most recent files are. The public domain arena for the Amiga is incredibly dynamic

and changing constantly with hundreds of new files and program appearing every month. You want to participate on a network that keeps up with these changes and is always able to deliver the very latest to you, as well as keep a stockpile of those "old standby" files and programs that every Amiga owner should have.

In other words, you wouldn't go out and buy a car without shopping around for the best model and price that suits your needs and budget. Do the same with online commercial networks. Make that network want to keep you as a customer. I think once you've tried Plink and the AmigaZone you'll like it well enough to want to stick around.

At this point, if you're interested in exploring Plink, you can get a trial signup kit in one of two ways:

Use your new modem and dial 1-800-826-8855, or if you live inside Illinois, dial 1-312-822-9712. When you are connected, hit your [RETURN] key a couple of times so that Plink will know what baud rate you're using, and then follow the instructions you will see on your screen to receive a trial subscription.

If you prefer, you can call by voice instead. Dial 1-800-524-0100 or if you live inside Illinois use 1-312-670-2666 and tell the customer service representative that you've read about Plink in this magazine and are interested in a trial subscription.

My Plink i.d. is "CBM*HARV" and once you've found your way to the AmigaZone, you'll see me in the club most every evening. Feel free to send me an online message and say hi, or leave me electronic mail. I'll be glad to help you in any way I can to make the online experience more enjoyable and productive for you.

Next time: Xmodem, Downloading, ARC, and related topics. ☐

About the author:

Harv Laser, CBM*HARV on Plink, is the Founder and Senior Chairman of the AmigaZone Club, and an Amiga owner since October, 1985. He lives in the Los Angeles area.

DEALERS ENQUIRIES

for ...



ATTENTION DEALERS!

Are you Interested in adding a high quality, fast moving publication to your line of Amiga products? How about **AmigoTimes!** We publish a monthly magazine that is accompanied by a disk packed full of currently available Public Domain programs and utilities, demos of commercial software, and programs written by our own programmers. This type of publication is a first in the Amiga community, and we are presenting you with the opportunity to be amongst the first dealers to carry **AmigoTimes** in your area (a definite edge over your competition). Everybody loves new, innovative, and visually impacting magazines. Your customers are your bread and butter, so it is in your best interest to keep them satisfied and on top of the latest Amiga developments. This is your opportunity to show your customers that you really care about keeping their business. Now is the time to order issues of **AmigoTimes** before you get a back log of requests.

FOR FURTHER
INFORMATION
CONTACT:

AmigoTimes

5124 St. Laurent, Suite 100
P.O.B. 1228
Ville de St. Catherine
Quebec J0L 1E0
CANADA
(514)638-6303

"C" Further

Use Execute() with parameter passing.....

It is often desirable to run one program from within another or to have several disk-based subroutines which can be called whenever they are required. This allows you to save memory while your program is running, and also lets programs run on machines which have less memory than the program itself. Being multi-tasking the Amiga is made to work in exactly that fashion; Workbench loads programs and libraries from disk when certain operations are done.

There are several ways to do this using 'C'. One way is to run the subroutine as a separate process, or it can be done by running it as a task. The above mentioned ways are the most effective and versatile if you want to do serious multi-tasking. The only problem is they are also the most difficult to use. There is one other way, and that is to use the AmigaDOS Execute() function. Many say this function is alright as long as you do not have to either pass parameters or receive a return value. This is not quite true, as the following two programs will demonstrate. I used the Execute() function to pass the address of a string from one program (master.c) to another (slave.c), the slave program then alters the string and exits. Yes I know, not very exciting, but it does show how you can pass values back and forth between programs by using the Execute() function. The function we use has the following format:

```
int success;          /* boolean, return value */
char *commandString; /* command name */
struct FileHandle *input, *output;
/* File Handle */
success = Execute(commandString, input, output);
```

The address of one or more

variables can be imbedded in the commandString using the sprintf function. In this way the address of a variable may be passed to another program, (ie. as a string). This is done in the following way:

Listing 1.

```
/* master.c */

char address[80] = "We have made it across !!!";

main()
{
    int error;
    char s[80];

    printf("MASTER: main started !");

    sprintf(s, "slave %ld", address);
    printf("MASTER: sending command: %s", s);
    error = Execute(s, 0, 0);
    if (error != NULL) {
        printf("Exiting: ERROR NUMBER = %d", error);
        exit(0);
    }
    printf("MASTER: string changed by slave %s",
           address);
    printf("MASTER: exiting !!");
    exit(0);
} /* end of main */
```

Once the command string is passed to the slave program it is then decoded into an address:

```
char *parameter;
/* ptr to the string from master.c */
long address; /* ptr value in numeric form */
address = atoi(argv[1]);
/* convert second argument to long */
parameter = (char *)address;
/* convert long to ptr of type char */
```

The second argument is converted from an ASCII string to a long value by the atoi() function. The long value is then converted to the value of a pointer of type 'char', now the slave program has the address of a string defined in the program master.c and can therefore

modify it (ie. pass back a return value). The next listing will show how the slave program handles its side of the communications:

Listing 2.

```
/* slave.c */

#include <stdio.h>
#include <exec/types.h>

main(argc, argv)
int argc;
char *argv[];
{
    int i;
    char *parameter;
    char s[80];
    long address;

    printf("SLAVE: Process working!");

    address = atoi(argv[1]);
    parameter = (char *)address;

    printf("Command String %s is", argv[0], argv[1]);
    printf("Address of parameter %ld", parameter);
    printf("Parameter String %s", parameter);

    sprintf(parameter, "The Slave was here!!");

    printf("SLAVE: Process closing!");

    exit(0);
} /* end of slave */
```

Remember, this is not the most versatile, or direct, way of running disk-based routines, but it is quick and it works. I hope you will find some occasion to put these routines to some use. For serious multi-tasking though, you should create a separate process, or task, and communicate with the tools supplied by the system.

These listings can also be found on the accompanying *AmigaTimes* disk. □

By Eyo Sama

GOBBLEDYGOOK

GOBBLEDYGOOK

GOBBLEDYGOOK

WARNING:

The material presented herein is not intended to offend readers or specific groups of individuals. Its purpose is intended purely to be ludicrous, insane, humorous, and/or full of lies (its your choice). Neither is the material presented here the opinion of this magazine, but those of its author; who is considered to be a bit too wild and/or has been told to seek psychiatric treatment. This column is not intended for mature adults. Those people considering themselves as mature adults and are reading this are welcomed converts. This column can't promise to be printed monthly due to censorship existing at all levels of this magazine. But on those occasions that I manage to distract, gag and tie (sometimes bribe) the managing editor, it will appear.

LETTERS

Dear AmigoTimes:

I just recently bought my external 3.5" drive for my A1000, but I am having problems getting it to use some of my left over 5.25" disks. What should I do?

Sincerely Ernest Nagy

Dear Mr. Nagy:

We suggest that you partially fold the 5.25" disks and with a wooden spoon force them into the drive. Making sure you use a wooden spoon, in that we have heard that metal spoons could damage your drive heads. Another solution would be to simply cut 1.75" off each side of your 5.25". Make sure you cut the correct two sides, this is easily determined with the flip of a coin. This solution is the simplest and no unnecessary forcing of disks are needed. P.S. don't feed your disk drives left overs. Wake up and smell the coffee!

CORRUPTED

Some of our staff have recently lost or had disks corrupted to such a great degree that Diskdoctor or Disksalvage could not repair or replace the damaged

files. So one of the staff members, trained in Voodoo, had to use Disk Xorcist to remove the Demons from the possessed disk. After which the disk returned to a normal disk wear state. Its possible this type of disk possession might be spread by telekinesis.

NEWEREST INDUSTRY

A new silicone chip manufacturer is attempting to break into the microchip industry with a radical new chip design. The old chip designs utilized "logic gates", says company president Silly I. Dea, to move data around. We have discovered a new data compression algorithm that will allow us to compress "Barn Doors" to occupy even less space than "Logic Gates" and still retain its characteristics. Presently the company has halted their spearhead research in order to find some semi-serious financial backing.

HARDWARISH

Some of our hardware experts are currently working on some innovative ideas on changing the chips in any Amiga computer to enhance their performance. Several companies are participating in this venture along with our staff. Some companies already involved are Humpty Dumpty, Ruffles, O'Gradies and Pringles. As soon as our staff finds the best chip combinations to replace we will notify you. Some problems already have arisen! Careful handling of chips are necessary because they can crumble easily and most of the supply is eaten while trying to decide how to go about the complicated task.

INSIDE AMIGOTIMES

The AmigoTimes staff was recently moved to its new location. The location can't be disclosed because we are chauffeur driven there, blind folded.

When we get to the office we are handcuffed to an amiga and strapped in to our chairs. Our work hours vary. They usually last the duration of finishing whatever task the Managing Editor (slave driver) gives us to do. Most of the staff goes home at reasonable slave work hours. But our Art director has not been able to finish his tasks on time lately, so he's being retained for 2 weeks straight. Boy I hope he finishes soon because we will be starting the next issue in a couple of days.

We have big windows on one side of the office, that has a great view of the concrete wall of a high rise 15ft from our building. Our boss always complains to us that we complain about our weekly salaries. He sighs the story about his dad who used to walk 50 miles to work before the sun rise, then working 25 hours a day and getting \$5.00/day (boy times have changed). He also says that we make more money than Rickshaw pullers in Calcutta, Ain't life wonderful. Roar, big freaking (can't say that copulating word) deal!

Anyway can't say much more about the other stuff, we are not allowed 2. Our boss says the Animal Cruelty officers will continue writing him nasty letters if we continue describing our work environment.

CONCLUSION

Well thanks for reading my Plea for Insanity and I hope I live through the next issue. So spread the news to all your friends about our staff situation and drop me a word on how you're doing. Hope you enjoyed this, cause depending on the response towards this column means if I get 40 lashes or not and weather I can continue writing it. Send in your vote now! Actually a petition would be appreciated. □

A QUICKER PACKER UPPER

An Assembly Language State Machine For IFF Image compression

Despite advances in processor and memory chip technology, I still believe faster is better. In the world of computers, no matter how fast something already is, it can generally be improved by making it faster yet. Since most routines written for the Amiga are in C or some other high-level language (Modula-2, Basic, etc.), you can usually speed them up by rewriting in 68000 assembly language.

This is what I have done with a critical image handling routine. Graphics on the Amiga almost universally abide by the "interchange file format" (IFF) specifications. And IFF images are almost universally stored in a special compressed form. A subroutine called PackRow() is used to perform that compression.

I will describe an assembler replacement for PackRow() - APACK.ASM - which is nearly three times faster and more than four times smaller than the "official" C version. Used in graphics software, it can speed up the image save function anywhere from two to nine seconds, depending on image type and resolution.

In addition, APACK.ASM provides a nice example of a very useful and fast software design tool called "the state machine". Even if you have no particular use for PackRow(), you may nonetheless be interested in the underlying conceptual toolkit.

EA IFF 85

The introduction of the Amiga coincided serendipitously with Electronic Arts' publication of their "EA IFF 85" stan-

dard for data files. The idea was to provide a sort of common language for microcomputers. By using the same algorithms and data formats, very different systems would nonetheless become "plug compatible", as it were. IFF would be to micros what MIDI is to electronic instruments.

Although presented as a technical specification, EA IFF 85 at times reads almost like an artistic or religious manifesto. The basic problem is how to represent information in a way that's program-independent, compiler-independent, machine-independent, and device-independent. The widespread adoption of the IFF standard by Amiga software has contributed greatly to the computer's preeminence as a system for sound and graphics. Electronic Arts deserves a lot of credit for this.

The part of the IFF spec in which we are interested concerns the storage of graphics. The raw data for a low resolution (320 by 200 pixel) 32 color image on the Amiga takes up 40,000 bytes: 5 bitplanes of 200 rows of 40 bytes each. A 16 color high resolution picture requires over three times that, or 128,000 bytes. That's quite a bit of data even for today's generous systems, so the IFF standard provides a protocol for compressing graphic images.

Compression is similar to searches and sorts in providing a playground for constantly evolving computational ingenuity. If you are interested in exploring it further, BYTE magazine in particular, has run a number of articles on the efficiency of different forms of compression (see Further Reading, below).

However, as a rule of thumb, the more efficient a compression algorithm (the greater the reduction in size), the more time it takes to pack and unpack the data.

cmpByteRun1 Compression

The standard IFF compression technique is a variant of something called "run length encoding". The protocol - cmpByteRun1 - represents a nice compromise between speed and efficiency. It reduces image data by more than 40% on average, yet only adds a few seconds to the time needed to save an image to file. As we shall see, by using PACK.ASM you can almost eliminate that time penalty altogether.

Following EA IFF 85, we can describe cmpByteRun1 by providing a routine that "decompresses" an image packed according to its protocol:

```
unpack()
{
    char *packed, *unpacked, control, byte;
    int i;

    while (MORE_DATA_IS_NEEDED)
    {
        control = getbyte(packed);
        if (control >= 0 && control <= 127)
            for (i = 0; i < control+1; i++)
                *unpacked++ = getbyte(packed);
        else if (control == -128)
            /* noop: do nothing */
        else /* if (control >= -127 && control <= -1) */
        {
            byte = getbyte(packed);
            for (i = 0; i < -control + 1; i++)
                *unpacked++ = byte;
        }
    }
}
```

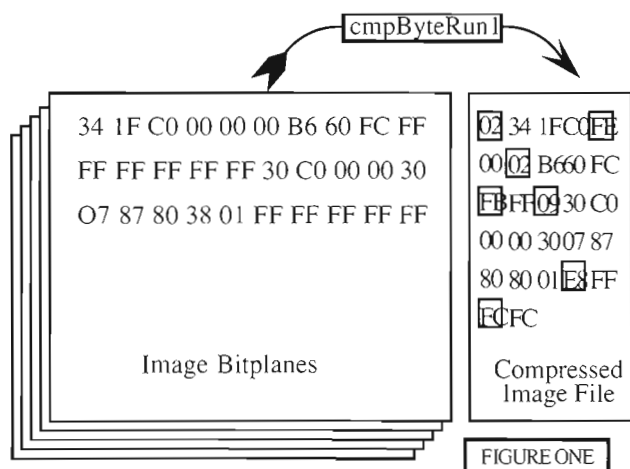
As you can see (assuming you can C), the compressed data is broken down into "runs": a "control" byte followed by some number of "data" bytes. If the

By Gerald Hull

control byte is positive (0 to 127), it indicates a "literal run": the next control+1 bytes should be taken into the uncompressed image as is. If the control byte is -128, it represents a special case - ignore it and get another control byte.

✱ If the control byte is otherwise less than zero (-1 to -127), this indicates a "replicate run": grab the following byte and put it into the uncompressed image -control+1 times consecutively.

Clearly the efficiency of cmpByteRun1 depends upon the image having significant areas colored the same. Things are a bit more complex than this in that the IFF standard requires "interleaved bit map" images (ILBMs) to be stored bitplane by bitplane. Thus, different byte values don't correspond directly to different colors, but instead one bitplane's component of the colors of eight successive horizontal pixels (Fig.1).



Still, the more monochromatic expanses in an image, the more likely it will contain the replicate runs that enable efficient cmpByteRun1 compression. On the other hand, for images that contain no or very few replicate runs, the cmpByteRun1 protocol can so be inefficient that it creates a "packed" image larger than the unpacked original.

This is because every literal run costs a byte: it takes n+1 bytes to pack n literal bytes. Conversely, every replicate run greater than 2 saves bytes: n-2 bytes, to be exact. Hence, in the worst case, if there are no replicate runs, it takes 41,000 bytes to "pack" a 40,000 byte low resolution, 5 bitplane image.

This is because ILBMs are stored

not only bitplane by bitplane, but row by row, and it takes 40 bytes to store a 320 pixel row. You might think the ideal solution would be to pre-scan an image, and compress only when it is efficient to do so. However, if execution speed is a priority, it is simpler to always use cmpByteRun1, knowing this usually will lead to greater efficiency.

A sample of 26 images on the DPaintII Art Disk suggests an average compression of 43% using that protocol. Consequently, virtually every graphics program on the Amiga automatically invokes cmpByteRun1 when saving IFF images.

Segmentation and Optimality

So far as the unpack() function is concerned, the cmpByteRun1 protocol simply requires a sequence of one or more literal or replicate runs. However, there is more to it than this. First, as we have

seen, an IFF ILBM image is stored and restored by row.

This means, for instance, that a sequence in a low resolution image couldn't contain a 25 byte literal run followed by a 16 byte replicate run. The runs must break down into segments of

exactly 40 bytes each.

Second, the code Electronic Arts has released into the public domain implementing their IFF spec requires a compressed image to meet a minimal requirement of optimality. You can use at most n+1 bytes to pack a row of n bytes, unless $n > 128$. For a row of any size n, a packing routine can generate at most $\text{MaxPackedSize} = n + ((n + 127) \gg 7)$ bytes. AMIGA MAIL, Commodore's "technical support publication" for developers, discusses this in the Sep/Oct 1987 issue:

The EA compressor expects a smartly compressed line, and will return an error if handed an encoded line

more than one control byte larger than [the] destination scan line.

Frankly, it's unclear to me why they regard this restriction as "smart". Software robustness is the capability to tolerate in-optimality, if not outright idiocy. And, in fact, many graphics programs have wisely eliminated it: Butcher 2.0 and Electronic Arts' own DPaintII, for example. Nonetheless, programs which use the EA public domain IFF routines "as is" will generate errors if you attempt to load a packed image which isn't minimally optimal.

The upshot, for our purposes, is that the algorithm for packing an image according to the IFF spec has to enforce that restriction. This, of course, imposes some price in terms of execution speed. The problem comes down to how you handle a replicate run of just two bytes. This is what EA IFF 85 recommends:

It's best to encode a 2 byte repeat run as a replicate run except when preceded and followed by a literal run, in which case it's best to merge the three into one literal run. Always encode 3 byte repeats as replicate runs.

When I read that, I was sure I could write a faster routine. If you check every 2 byte repeat to make sure it's not both preceded and followed by a literal run, you're going to waste a lot of processing time. My routine would simply ignore 2 byte repeats: replicate runs would start with 3 byte repeats.

Further, my routine would look for replicate runs; any bytes preceding one would automatically constitute a literal run. Consequently, I would never have two consecutive literal runs. Since my replicates start with 3 byte repeats, I would save a byte with every one. This guarantees I would meet the minimal optimality requirement. Every literal run but one, would be matched with a replicate run that would compensate for its extra byte.

Real-Time Programming

I had another trick up my sleeve. Much of my programming experience has been in the realm of real-time machine control and robotic intelligence. In this

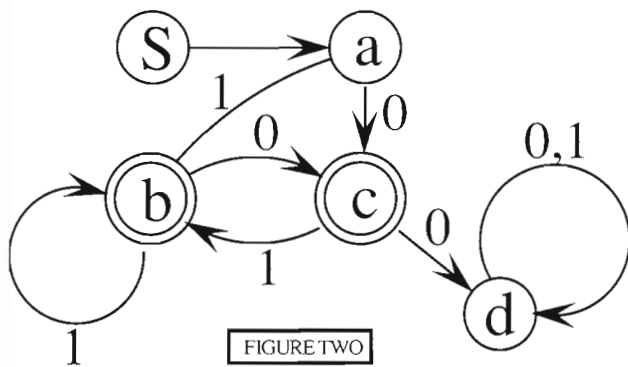


FIGURE TWO

field I have learned the great usefulness of state machines in the design of speedy algorithms. Virtually every real-time program I have ever seen is constructed on the state machine model, usually involving whole constellations of state machines. My slogan: "Real-time programmers use state machines."

Without going into mathematical complexities, you can regard a state machine as anything whose behavior is a function of the state it's in. Different types of input cause the machine to change from one state to another. You can get a good intuitive feeling for the concept through "state machine diagrams" (Fig.2).

Different states are represented by circles (called "nodes"), and labeled arrows ("directed edges") show how input causes a transition from one state to the next. The state machine in Figure 2 has four distinct states, identified with the lower case letters 'a', 'b', 'c', and 'd', plus a special "start" state shown with a capital 'S'. Starting in state a, if a 1 is input, it goes into state b; while if 0 is input, it goes into state c; and so forth. Two of the states, b and c, are final states, as shown by doubled circles. If

an input string of 1's and 0's causes the state machine to end up in a final state, it is said to "accept" that string.

So far, it may seem as if state machines are a perfect example of the boring stuff we are forced to memorize in mandatory computer math courses. Their

enormous versatility in software design will be a bit more apparent when you look at the examples in Figures 3 and 4. They reveal how such diverse systems as a soda machine and a video game are easily and perceptively represented in state machine terms.

The utility of state machines in programming may be more obvious when you realize they are implicit in the flow charts traditionally used to diagram software. There is a more theoretical justification for using state machines in data compression algorithms. It has to do with the fact that `cmpByteRun1` can be rendered as a "regular expression." In computation theory, a state machine is nothing more nor less than a regular expression parser. (Hopcroft and Ullman address this at exquisite length; see Further Readings.)

State Machines in C

Most high-level languages include control structures which directly facilitate the implementation of state machines: a case statement inside a repeat loop. In C, the basic format looks like this:

```
while (INPUT_CONTINUES)
{
    switch (state)
    {
```

```
        case 0:
            /* code for state 0 */
            break;
        case 1:
            /* code for state 1 */
            break;

        /* however many more states as you need */
    }
}
```

Logically speaking, the case statement can always be replaced by a concatenated sequence of if-then-elses. However, the latter imposes a much higher price in terms of execution speed. Conditional branching almost always involves a greater time penalty. However, be aware that some C compilers sometimes downcode switch statements into multiple conditionals.

So the speed advantage of the state machine consists in the ability to "jump" right to where you want to go, the way express lanes in supermarkets are supposed to work. This is why real-time programmers use state machines. For a more extended C example, here's how you could implement the state machine in Figure 2:

```
#define ERROR -1
#define ACCEPT 0
#define DONT_ACCEPT 1
#define ZERO '0'
#define ONE '1'

state_machine(input)
char *input;

{
    char i, state = 'a';

    while ((i = *input++) != '\0')
    {
        switch (state)
        {
            case 'a':
                if (i == ZERO)
                    state = 'c';
                else if (i == ONE)
                    state = 'b';
                break;
        }
    }
}
```

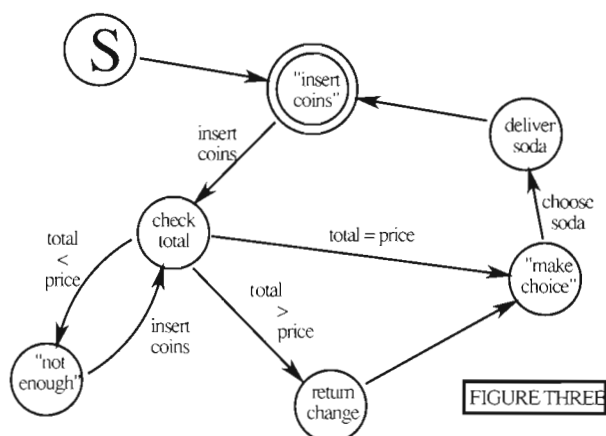


FIGURE THREE

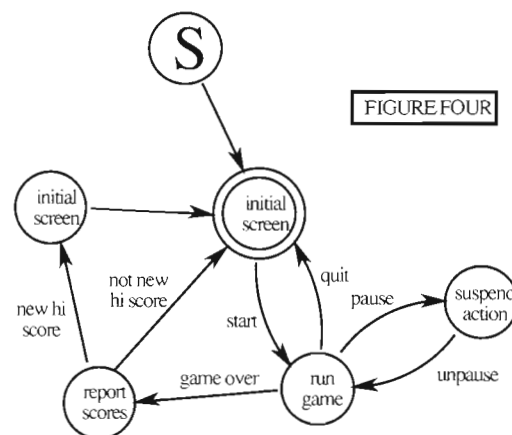


FIGURE FOUR

```

case 'b':
  if (i == ZERO)
    state = 'c';
  else if (i == ONE)
    state = 'b';
  break;

case 'c':
  if (i == ZERO)
    state = 'd';
  else if (i == ONE)
    state = 'b';
  break;

case 'd':
  if (i == ZERO)
    state = 'd';
  else if (i == ONE)
    state = 'd';
  break;

default:
  /*** ERRONEOUS INPUT ***/
  return(ERROR);
}

```

```

if (state == 'b' || state == 'c')
  return(ACCEPT);
else return(DONT_ACCEPT);
}

```

In assembler the speed advantage of a state machine is even greater. There are different forms of implementation, but they all provide a way to jump directly to the code implementing the current state without multiple tests and conditionals. They save time over the high-level approach by branching immediately back to the top of the repeat structure, instead of first down to the bottom.

Here are two different ways to implement them in 68000 assembly language:

```

CAS    equir    d0
STA    equir    d1

clr     STA

REPEAT
  move   CAS,STA
  asl    #1,STA
  jmp    TABL(PC,STA)

TABL    bra     CASE0
        bra     CASE1

* however many states as you want

CASE0
* code for state 0
  bra    REPEAT

CASE1
* code for state 1
  bra    REPEAT

```

The second way takes a little longer and imposes additional register and memory expenses, but allows you to render the jump table as a simple list of longword addresses.

```

CAS    equir    d0
STA    equir    d1
OFF    equir    a0

```

```

clr     STA

REPEAT
  move   CAS,STA
  asl    #2,STA
  move.l TABL(PC,STA),OFF
  jmp    (OFF)

TABL    dc.l     CASE0
        dc.l     CASE1

* et cetera, et cetera

CASE0
* code for state 0
  bra    REPEAT

CASE1
* code for state 1
  bra    REPEAT

```

Further Optimization

The advantage of maintaining the current state in a separate variable is that you can monitor the flow of execution by keeping track of what state you are in. An extremely useful diagnostic tool for real-time programming is a screen that constantly updates the current state of all the component state machines.

However, for small programs, once you are quite certain you've gotten all the bugs out, you can have the different states branch directly to each other. And if the next state is the succeeding state, you can eliminate the branch altogether, and simply "fall through" into it. This eliminates all the processing required to maintain and utilize the state variable.

If you look at the listing of APACK.ASM, you can see the results of this simplification. I must emphasize that it is difficult to debug, and almost impossible to maintain, this "abbreviated" kind of assembler state machine code. In fact, it is extremely difficult to distinguish from the "Bowl of Spaghetti" (BS) code every structured programming course warns you against (Fig.5).

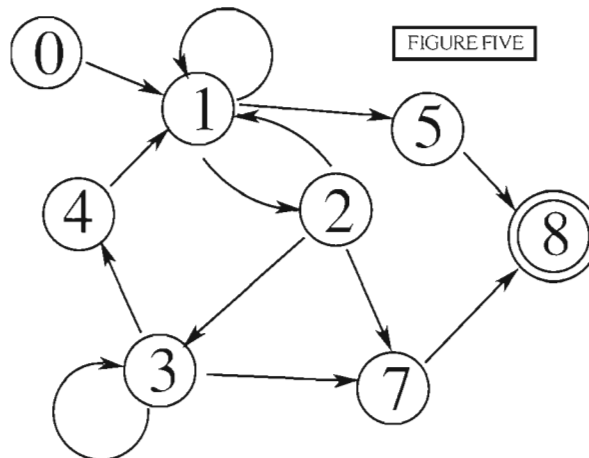


FIGURE FIVE

Comparison with PACKER.C

When I finally took time to study Electronic Arts' version of the image compression routine, I discovered some of my assumptions were unfounded. It doesn't check for the case of a two byte replicate run sandwiched by two literal runs, in order to reprocess it as a single long literal run. In fact, in terms of efficiency, EA's PACKER.C employs a cleverer algorithm. It takes advantage of the fact that 2 byte repeats immediately preceded by a replicate run can always be treated as replicates.

As a result, the compression achieved by PACKER.C is slightly better than that of APACK.ASM. In addition, I discovered that the EA version checks to make sure runs do not exceed the 128 byte limit. (128 is, of course, the maximum positive number expressible in a single byte.) I had simply ignored this possibility, since packing is done on a per row basis. At eight pixels per byte, that implies screens with a horizontal resolution greater than 1024 pixels. So far as I know, such resolutions are not yet on the Amiga horizon.

Finally, PACKER.C is also built on the state machine model, although only rudimentary in that it distinguishes only two states (in contrast to the eight of APACK.ASM). However, the Electronic Arts code is targeted for all micros, not just Amigas or even 68000 based machines. This imposes not only the inherent speed limitations of high-level language, but also a number of checks and considerations irrelevant to our purposes.

For those reasons, and not just because of clumsy C programming techniques, APACK.ASM is much smaller and faster than the high-level routine it replaces. The speed improvement is shown in Table 1.

There, four different versions of PACKROW are pitted against each other: APACK.ASM, APACK.ASM with checking that makes it 1024 + pixels-per-row compatible, PACKER.C, and

TABLE ONE: COMPARISONS OF cmpByteRun1 PACKING SPEED

	IMAGE 1	IMAGE 2	IMAGE 3	IMAGE 4
raw size	40,000	40,000	40,000	40,000
NOPACK.C				
comp size	40,000	40,000	40,000	40,000
seconds	1.0	1.0	1.0	1.0
APACK.ASM (no chk)				
comp size	40,177	36,009	18,808	7094
seconds	1.5	1.4	1.2	1.1
APACK.ASM (check)				
comp size	40,177	36,009	18,808	7094
seconds	1.7	1.6	1.3	1.2
PACKER.C				
comp size	40,177	36,004	18,006	7094
seconds	4.2	4.1	3.4	2.9
improvement over PACKER.C:				
NOPACK.C				
factor	4.5	4.3	3.6	3.1
seconds	3.3	3.1	2.5	2.0
APACK.ASM (no chk)				
factor	2.8	2.9	2.9	2.7
seconds	2.7	2.7	2.2	1.9
APACK.ASM (check)				
factor	2.6	2.6	2.6	2.4
seconds	2.6	2.5	2.1	1.7

NOTES: Each of the images tested is low resolution (320 by 200) with 5 bitplanes (up to 32 colors). All saves are to RAM;; other devices will take longer. Apparent anomalies in the math are the result of rounding.

NOPACK.C; the last is a routine that outputs the raw image, that is, without compression.

Four different 320 by 200 low resolution five bitplane images of differing degrees of cmpByteRun1 compressibility measure the relative capabilities of those routines. The time in each case reflects an actual save to RAM:. Since these timings are a direct function of image size, it's easy to calculate the improvement for other types of ILBM images. A four bitplane 640 by 400 high resolution picture, for example, would increase the "savings" by a factor of $4/5 * 4 = 3.2$.

In the speed derby, the routine which outputs the image without compression, NOPACK.C, is the winner in every instance. This is despite the fact that its output file is sometimes more than four times larger. However, APACK.ASM without checks comes in close behind, at worst a half-second slower in the examples given. Adding checks slows it down only a couple tenths of a second more.

APACK.ASM

The bottom line is that APACK.ASM can perform the same PackRow() function an average of 2.8 times faster than PACKER.C, with only a tiny loss in compression efficiency (99.993 %). It has the same parameter list as PACKER.C, which means you can link the 200 (or 248) byte PACK.O into your code in place of the 856 byte PACKER.O, and get exactly the same results. However, please note the disclaimer in Listing One. If you want to guarantee your code will work with resolutions as yet (to me) unborn and unnamed, the average speed-up factor is still nearly 2.6.

The state machine diagram for APACK.ASM is given in Figure 5. I haven't tried to fill in the inputs and nodes: you can easily infer them from the comments in Listing 1. I will refrain as well from a line by line explanation of the code. If you are interested, you will enjoy figuring it out for yourself.

Some general comments, however, should help illuminate what's going on. Three pointers are maintained: IP, PT,

and IQ. Between IP and PT is a literal run, if any; between PT and IQ is a replicate run, if any.

```

***** input stream *****
      ^           ^           ^
      IP <lit run> PT <rep run> IQ

```

Whenever input terminates or a replicate run comes to an end (or MAX is reached, if you are using CHECK), any runs that have built up are "flushed" to the destination buffer.

APACK.ASM may seem a bit redundant because I avoid conditional tests whenever possible, even when that requires code repetition. However, APACK.O is still 3.5 to 4.3 times smaller than PACKER.O. You could simplify the code's appearance with some macros, but it's easier to understand all written out. However, remember that APACK.ASM is highly optimized, and things aren't always what (or why) they seem.

Further Reading

"A Better Way to Compress Images." Michael Barnesley and Alan Sloan, BYTE, January 1988.

"Data Compression with Huffman Coding." Jonathan Amsterdam, BYTE, May 1986.

EA IFF 85. The March 25, 1987 release is available as Fred Fish Disk 64. Later releases can be obtained from CATS-Orders, 1200 Wilson Drive, West Chester, PA 19380 for \$20.00 (Canada add \$2.50 shipping, all others add \$5.00).

"Focus on Algorithms: Run-Length Encoding." Dick Pountain, BYTE, June 1987.

"INTRODUCTION TO AUTOMATA THEORY, LANGUAGES, AND COMPUTATION". Hopcroft and Ullman, Addison-Wesley, 1979. If you can tolerate the math, you can learn an awful lot from this book about state machines and their powerful cousins. □

(due to space limitations APACK.ASM was not printed within the magazine but has been included on the disk. ED).

Modula-2 File Requester

In this article, the first of two of a series, I will show you how to write a file requester in Modula-2. A file requester is a small window for interactively selecting a file name. If the user of this program knows Modula or even Pascal, he/she should be able to follow along. I will try to explain everything that has to do with Amiga specific items in great length. The source code for this article has been written to be compiled with the Benchmark Modula-2 compiler.

One thing that the 'Amiga Modula-2 world' has been missing is a file requester. I've been waiting a long time for one to show up on the networks, but have seen nothing. So, one time when I was trying to figure out what project to do next, I figured I might as well buckle under and do it myself. At first glance, it looked like a huge program. But as every good programmer knows, huge programs are just a bunch of small ones.

Where to start? The first thing that must be done, when starting a program of this size, is to break it up into parts. At first glance, I see two big parts. The first part, discussed this month, is to read directories from disk. The second part is to provide a user interface.

Modula-2 is the perfect language for this modular approach. We can write a procedure to read directories and put it in its own library module. Once debugged and compiled we'll never need to edit or compile this part of the program again. Our interface module (.def) provides all of the external information we'll need so that we don't have to keep going back to the source code. The interface module also helps us to keep our modular approach. Once we define the interface, we can code each module without having to know what the other modules will contain.

In order to write a procedure in a separate Module to read directories, we must first design a data structure that our Modules will use to pass information back and forth. The data structure holds information about all of the entries in the directory. There are three things to know about each directory entry; the entry name, the file size and whether the entry is a file or a directory. Here is the data structure:

```
FileEntry = RECORD
  Name : ARRAY [0..30] OF CHAR;
  Type : BOOLEAN;
  Size : LONGINT;
END;
```

Since the data structure is used to hold information about many directory entries, we will use an ARRAY of FileEntry's and define the data area like this:

```
List : ARRAY [0..200] OF FileEntry;
```

This array provides slots to hold information for about 200 directory entries. Slot zero is reserved for future use.

DEFINITION MODULE

Now it's time to start coding. Because we are going to put our procedure in a library module, the first thing we must do is create a definition module. We've already written the definition of our data structure, so the only thing left to write for the definition module is our procedure declaration. Three parameters will be used. First, we must pass our array of FileEntry's. Then pass the length of the array and, finally, we need to pass the path of the directory to be read. This procedure also returns, to the calling program, the number of directory entries read. Here is our procedure declaration:

```
PROCEDURE ReadDir (VAR List : ARRAY OF FileInfo;
  Count : CARDINAL;
  FilePath : ARRAY OF CHAR): INTEGER;
```

Listing 1 shows what the completed DEFINITION MODULE looks like.

READING DIRECTORIES

To read directories, we'll use four AmigaDOS procedures.

Lock() - This procedure 'locks' a file or directory. Locking a file tells AmigaDOS that we are using a file or directory and also provides a way to reference a file when talking to the operating system.

Unlock() - This procedure 'unlocks' a file or directory.

Examine() - This procedure is used to get information about a directory or file.

ExNext() - This procedure is used to get information about each directory entry in a directory.

Examine() and ExNext() use a data structure called FileInfoBlock. This is what the FileInfoBlock data structure looks like:

```
FileInfoBlock = RECORD
  fibDiskKey : LONGINT;
  fibDirEntryType : LONGINT;
  fibFileName : ARRAY [0..107] OF CHAR;
  fibProtection : ProtectionSet;
  fibEntryType : LONGINT;
  fibSize : LONGINT;
  fibNumBlocks : LONGINT;
  fibDate : DateStampRecord;
  fibComment : ARRAY [0..115] OF CHAR;
END;
```

We'll use three of the items in this data structure; 'fibSize' holds the size of the file in bytes, the first 30 characters of fibFileName holds the file name and

By Mike Scalora

fibDirEntryType indicates whether the entry is a file or directory.

To read a directory you have to: Lock() the directory, Examine() the FileLock, ExNext() the FileLock for each directory entry and then Unlock() the FileLock. ExNext() returns information about each directory entry in the FileInfoBlock. This is the pseudo-code that reads directories in AmigaDOS:

```
Lock() a directory
Examine() the directory
repeat
    ExNext()
until done
Unlock() the directory
```

GETTING DOWN TO BUSINESS

Now we start the real coding. The first thing to do is declare a few variables. We'll need four variables (I told you this would be easy!). To begin with, we need a variable of type FileLock (that we import from the AmigaDOS Module). Next, we need a few CARDINAL variables for counters and indexes. Lastly, a FileInfoBlock.

This is where things get a little sticky. Because of the way AmigaDOS was written, FileInfoBlocks have to be longword aligned. 'Longword aligned' means that the starting address must be divisible by four. We can't just declare a FileInfoBlock as a variable because Modula-2 does not assure us that all variables will be longword aligned. We are told however, that any memory allocated from the operating system will be longword aligned. So, instead of declaring a FileInfoBlock, we'll do the next best thing and declare a FileInfoBlockPtr. Then, just allocate our FileInfoBlock by using the ALLOCATE() procedure in the Storage library module.

Here's what our variable declarations will look like:

```
VAR
MyLock      : FileLock;
C, Num      : CARDINAL;
MyInfo      : FileInfoBlockPtr;
```

A CLEAN SWEEP

The next thing to write is a procedure to de-allocate any system resources that will be used. ReadDir() uses two system

resources. We will be allocating memory for a FileInfoBlock and we will also be using a FileLock. To make sure we don't de-allocate anything we didn't allocate, we'll use an IF statement for each resource. Here is the CleanUp procedure:

```
PROCEDURE CleanUp;
BEGIN
    IF MyLock#FileLock(0) THEN Unlock(MyLock); END;
    IF MyInfo#NIL THEN
        DEALLOCATE(MyInfo, SIZE(MyInfo^)); END;
END CleanUp;
```

Now, so that the CleanUp procedure works properly, the first two lines of our program are:

```
MyLock := FileLock(0);
MyInfo := NIL;
```

By setting these variables to "zero" values, the IF statements in CleanUp can tell if the resource has been allocated or not. The method of using a CleanUp procedure becomes more and more valuable as the size of the programs increase.

DIPPING INTO THE WELL

Now, we will write the code needed to allocate memory for our FileInfoBlock. Benchmark Modula-2 provides the procedures ALLOCATE() and DEALLOCATE() for our use. Some other compilers use NEW() and DISPOSE() - they do about the same thing. But one big difference is that ALLOCATE() needs to know how much memory you need to allocate. The SIZE() built-in function makes this simple. One thing to be careful of when using the SIZE() function and pointers is to make sure you put the caret (^) on the end of the pointer name. SIZE(MyInfo) would return 4, the size of the pointer, but SIZE(MyInfo^) returns 258, the size of the FileInfoBlock data structure that MyInfo points to.

```
ALLOCATE(MyInfo, SIZE(MyInfo^));
IF MyInfo#NIL THEN CleanUp; RETURN(-1); END;
```

The IF statement checks to make sure that the operating system really did give us memory. If our pointer equals NIL, then we know that the operating system did not give us memory for one reason or another, so we CleanUp and return an error code of -1. You will see

a lot of IF statements that look like:

```
IF condition THEN CleanUp; RETURN(-1); END;
```

These IF statements perform error checking after each operating system call. These IF statements prevent what is called, writing through a NIL pointer. Most operating system calls return zero, a negative value or a NIL/NULL pointer to indicate that there was an error. You MUST perform this error checking. Many bugs having to do with writing through a NIL pointer, only show up as GURUs when some type of system resource, like memory, is used up. For example, if we didn't do the error checking after the ALLOCATE() call because the system was out of memory, we would meet the guru as soon as we made our Examine() call. Since this file requester will be used in other programs that'll have problems of their own, we want to make extra sure that our code is bullet-proof.

THE BIG LOCKUP

Now we will try locking the directory. To us, a FileLock is just a number that we use to talk about a file with the operating system. To the system, a FileLock is a special kind of pointer that points to a FileLockRecord. Just as with the ALLOCATE(), Lock() uses a 'zero' value to indicate that we did not get a lock, so we have an IF to make sure things are going OK.

```
MyLock := Lock(ADR(FilePath), SharedLock);
IF MyLock#FileLock(0) THEN CleanUp; RETURN(-1);
END;
```

Next, we have to Examine() the directory. Examine() will then return a BOOLEAN value of TRUE or FALSE to tell you if there was an error. We pass Examine() our FileLock and the FileInfoBlock that we allocated.

```
IF NOT(Examine(MyLock, MyInfo^)) THEN CleanUp;
RETURN(-1); END;
```

If Examine() returned TRUE, we know that the operating system has filled our FileInfoBlock data structure with information about the file or directory that we have the lock on. This should be a directory, specifically, the directory that we are to read. To make

sure that it is a directory, we use and IF statement. Even if Examine() worked, we might have examined a file and not a directory, so in case our FileLock is on a file, we need an IF statement that looks something like this:

```
IF MyInfo^.fibDirEntryType<0D THEN CleanUp;
RETURN(0); END;
```

If MyInfo^.fibDirEntryType is less than zero, it's a file, if MyInfo^.fibDirEntryType is greater than zero, it's a directory. If it's a file, we return a zero telling the calling program that we were not able to read any directory entries, but there were no errors.

MAIN STREET

We are just about to get to the main loop of the program. But first we'll do a little housekeeping. To keep track of how many directory entries we've read, we need to keep a counter. Here is where we have to initialize the counter.

```
Num := 0;
```

Now, for our main loop we will use a WHILE. In the WHILE statement's condition, we will AND two things together. First, we'll use the ExNext() call itself. If ExNext() returns FALSE, we are done reading the directory. To make sure that the calling program's array is large enough, we'll check our variable 'Num' against the size of the calling program's array that is in 'Count'. All put together, here is the WHILE statement:

```
WHILE ExNext(MyLock,MyInfo^) AND (Num<Count) DO
```

If the ExNext() procedure returned TRUE, and we know it did because we are inside the WHILE loop, MyInfo^ now has information about a directory entry. Inside the loop, the first thing we will do is increment our counter of the number of files we have read.

```
INC(Num);
```

Now we need to write the code to move the information we want from MyInfo^ into the calling programs array. For the file name, we will use a quick FOR loop. For the entry type, we use a boolean expression to set it to

TRUE or FALSE. For the file size, a simple assignment will do. Here is what all that code looks like:

```
FOR C := 0 TO 29 DO
  List[Num].Name[C] :=
    MyInfo^.fibFileName[C];
END;
List[Num].Name[30] := 0C;
List[Num].Type := MyInfo^.fibDirEntryType<0D;
ELSE List[Num].Type := FALSE; END;
List[Num].Size := MyInfo^.fibSize;
END;
```

As you noticed, we also ended the WHILE loop. You did notice didn't you?

All that is left, is to cleanup and return the number of directory entries read to the calling program. I bet by now you could do this blindfolded!

```
CleanUp;
RETURN(INTEGER(Num));
```

Only the hardest part is left now. Yep, you guessed it. The old empty module initialization section!

```
BEGIN
END ReadDir;
```

Listing 2 shows what it all looks like put together. See, I told you it was simple. What good is a program that you can't use? Well, I was thinking that myself, so I whipped up a little program that uses ReadDir() to read a directory and then prints out the directory list to the screen. To use ReadDirTest, just type:

```
ReadDirTest DF0:
```

You should get a listing of all the files on the root directory of the disk in drive 0 (DF0:).

IN MY CRYSTAL BALL

Next month we'll start working on the user interface. It's much more complex and I'm going to try to move a little faster, but as long as we keep to our step by step approach, we should have no trouble.

For more detailed information about how to AmigaDOS library functions, or procedures as we call them, see Programmer's Guide to the Amiga by Robert Peck from SYBEX. Chapter 3 of the book has a section called 'AmigaDOS Directory Structure' which ex-

plains in detail most of the subjects touched upon in this article. Chapters 4 and 5 would be good reading for rest of this programming project. Robert has told me that Modula-2 source code is available for all of the program examples in his book on disk. You can use the order blank in the back of the book, just make sure you include which compiler you have with your order.

If you have any comments about this article and program or anything else about Modula-2, I can be reached at PeopleLink id SCARY, BitNet id SCALOR01@SNYNEWBA, CIS id 73717.2334 or by mail care of this magazine.

Until next month, may Wirth be with you!

Listing 1

```
DEFINITION MODULE ReadDir;
(* (C) 1988 Mike Scalora *)
(*****)
(*      ReadDir 1.0      *)
(* - ] Mike SCARY Scalora [ - Plink: SCARY *)
(*      *)
(* This MODULE is freely distributable for *)
(* noncommercial use as long as this notice *)
(* stays in. *)
(*      *)
(*****)

TYPE
(* this is the RECORD to keep the file info *)
  FileEntry = RECORD
    Name : ARRAY [0..30] OF CHAR;
    Type : BOOLEAN;
    Size : LONGINT;
  END;

PROCEDURE ReadDir(VAR List : ARRAY OF FileEntry;
  Count : CARDINAL;
  FilePath : ARRAY OF CHAR
): INTEGER;

END ReadDir.
```

Listing 2

```
IMPLEMENTATION MODULE ReadDir;
(* Copyright 1988 Mike Scalora *)
(*****)
(*      ReadDir 1.0      *)
(* - ] Mike SCARY Scalora [ - Plink: SCARY *)
(*      *)
(* This MODULE is freely distributable for *)
(* noncommercial use as long as this notice *)
(* stays in. *)
(*      *)
(*****)
(* 1.0 02-09-88 Mike Scalora *)
(*****)

FROM SYSTEM IMPORT BYTE, ADR, ADDRESS, SHIFT;
FROM AmigaDOS IMPORT FileLock, FileInfoBlockPtr,
  FileInfoBlock, Lock, Unlock,
  Examine, ExNext, SharedLock;
FROM Storage IMPORT ALLOCATE, DEALLOCATE;

(*****)
(*      ReadDir PROCEDURE      *)
(*****)

PROCEDURE ReadDir(VAR List : ARRAY OF FileEntry;
  Count : CARDINAL;
  FilePath : ARRAY OF CHAR
): INTEGER;
```

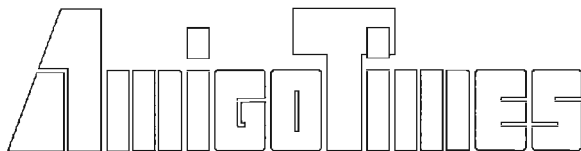
ADVERTISERS

Special Note to Advertisers:

AmigoTimes is a full featured color magazine accompanied by a disk. This allows our readers to experience the best of both worlds; an easily readable media without the need for a computer and a 3.5" diskette that contains PD software, demo programs from commercial software producers, along with program listings from the magazines authors.

Advertising in AmigoTimes will allow you to reach a broader range of customers while at the same time allowing them to have hands on experience with demo versions of your commercially available software. We are contributing the first Magazine & Disk publication to the exciting and rapidly expanding Amiga market. Within the next 6 months the Amiga market will expand rapidly, this translates into increasing sales.

This is the time to call AmigoTimes and reserve advertising space at competitive rates in a high quality publication. AmigoTimes accepts color and B&W ads in several standard page sizes. AmigoTimes also encourages new developers to get in touch with us.



CONTACT:

Lisa Sama
AmigoTimes
 5124 St. Laurent, Suite 100
 P.O.B. 1228
 Ville de St. Catherine, Quebec.
 CANADA J0L 1E0
 (514)638-6303

```

VAR
  MyLock   : FileLock;
  C,Num    : CARDINAL;
  MyInfo   : FileInfoBlockPtr;

PROCEDURE Cleanup;
(* ReadDir's own little Cleanup! How cute! *)
BEGIN
  IF MyLock#FileLock(0) THEN
    Unlock(MyLock);
  END;
  IF MyInfo#NIL THEN
    DEALLOCATE(MyInfo,
               SIZE(MyInfo^));
  END;
END Cleanup;

BEGIN
(* these inits assure that Cleanup; works *)
  MyLock := FileLock(0);
  MyInfo := NIL;

  ALLOCATE(MyInfo,SIZE(MyInfo^));
  IF MyInfo#NIL THEN
    Cleanup; RETURN(-1); END;

  MyLock := Lock(ADR(FilePath),SharedLock);
  IF MyLock#FileLock(0) THEN
    Cleanup; RETURN(-1); END;

  IF NOT(Examine(MyLock,MyInfo^)) THEN
    Cleanup; RETURN(-1); END;
(* it didn't work, abort *)

  IF MyInfo^.fibDirEntryType<0D THEN
    Cleanup; RETURN(0); END;

(* it must be a dir! *)
  Num := 0;
  WHILE ExNext(MyLock,MyInfo^,1) AND (Num<Count) DO
    INC(Num);
    FOR C := 0 TO 29 DO
      List[Num].Name[C] :=
        MyInfo^.fibFileName[C];
    END;
    List[Num].Name[30] := 0C;
    List[Num].Type := MyInfo^.fibDirEntryType<0D;
    List[Num].Size := MyInfo^.fibSize;
  END;

  Cleanup;
  RETURN(INTEGER(Num));
END ReadDir;

BEGIN (* no init code *)

END ReadDir.

```

Listing 3

```

MODULE ReadDirTest;
(* Public Domain 1988 Mike Scalora *)

FROM ReadDir   IMPORT ReadDir, FileEntry;
FROM System    IMPORT argv, argc;
FROM Terminal  IMPORT WriteString, WriteLn;
FROM LongInOut IMPORT WriteLongInt;
FROM Storage   IMPORT ALLOCATE, DEALLOCATE;

VAR
  MyDir : POINTER TO ARRAY [0..200]
    OF FileEntry;
  Num,I : INTEGER;

BEGIN
  IF argc<2 THEN HALT; END;
  ALLOCATE(MyDir,SIZE(MyDir^));
  IF MyDir#NIL THEN HALT; END;

  Num := ReadDir(MyDir^,200,argv^[1]^);

  IF Num>0 THEN
    FOR I := 1 TO Num DO
      IF MyDir^[I].Type THEN
        WriteLongInt(MyDir^[I].Size,10);
      ELSE
        WriteString(' -- dir --');
      END;
      WriteString(' ');
      WriteString(MyDir^[I].Name);
      WriteLn;
    END;
  END;

  DEALLOCATE(MyDir,SIZE(MyDir^));
END ReadDirTest.

```



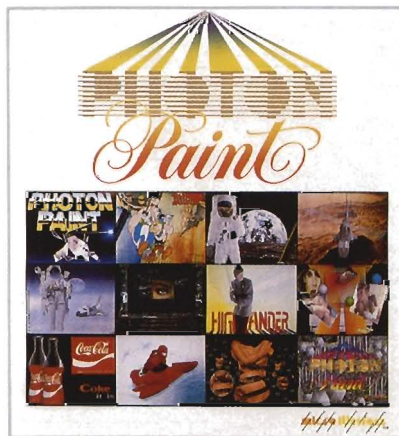


EXPANDING YOUR PAINT CAPABILITIES AS NEVER BEFORE

PHOTON PAINT IS A POWERFUL LO- AND HI-RESOLUTION HOLD AND MODIFY (HAM) PAINT PROGRAM, WITH OVERSCAN (NTSC /PAL) AND SPECIAL EFFECTS FOR THE AMIGA COMPUTER!

Now many of the features and techniques previously only available in separate Amiga paint programs have been combined and expanded in PHOTON PAINT! These include features that were limited to either a 32 color paint program, or a HAM paint program. Yet it doesn't stop there! PHOTON PAINT expands your paint capabilities as never before with these features:

- 64 color palette with 4,096 color alterations
- State of the art Surface Mapping •
- Exceptionally sophisticated brush operations



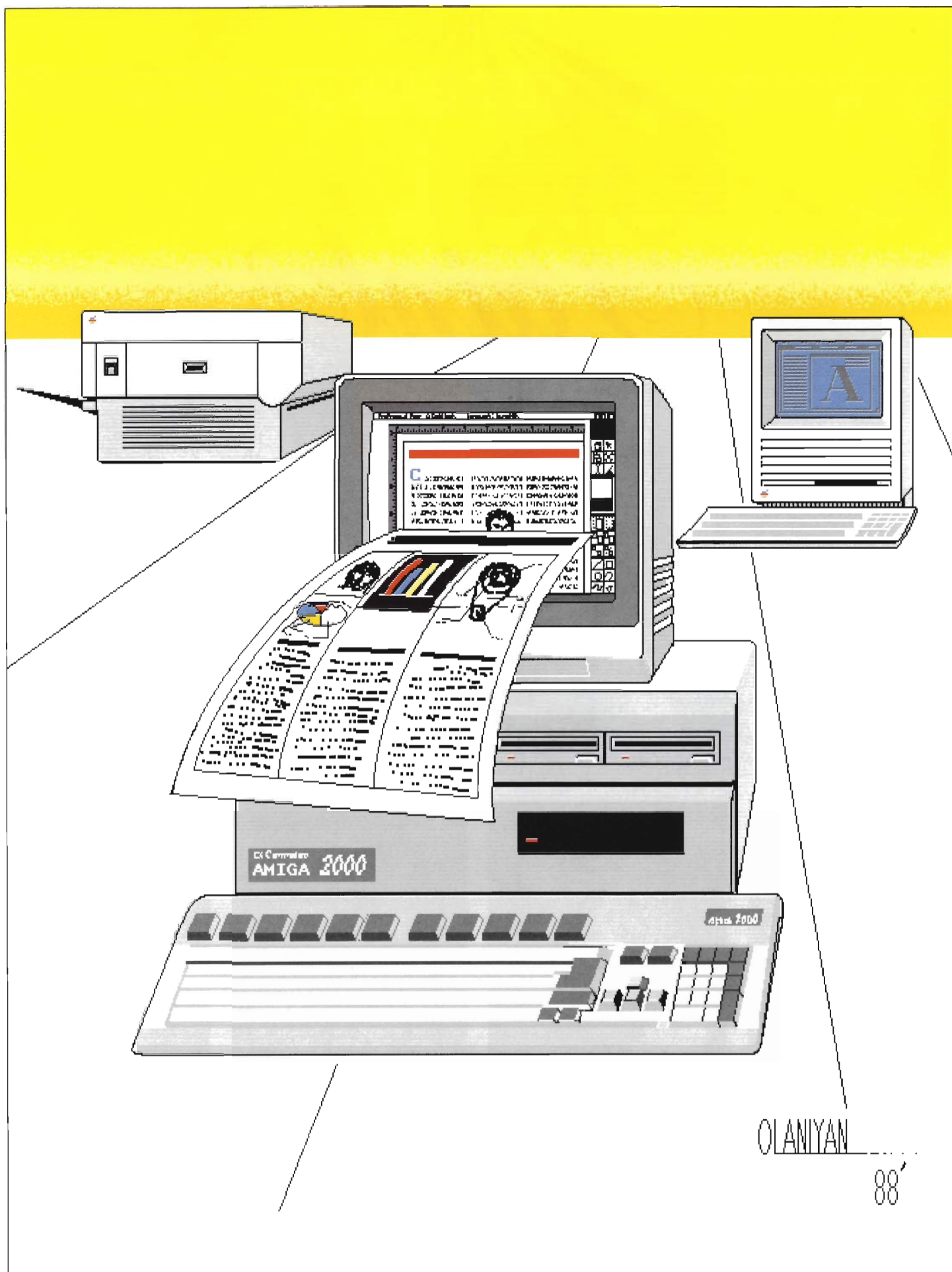
- Real-time operation of drawing tools • Paint in exactly the color you choose in HAM mode
- Allow HAM to modify colors • Two types of fill functions • All popular brush operations • Full blend mode • Total menu movability with brush and color palette menu size reduction option • Light source specification, normal, intense or dithered • Unique magnification window with variable pixel sizing • Transparent or opaque background colors • Palette color spreads with both (RGB) AND (HSV) functions
- Compatible with most third party art and animation systems, including Photon Video.

**EXPAND YOUR CREATIVITY TO
A NEW DIMENSION, PICK UP
PHOTON PAINT TODAY!**



microillusion™

17408 Chatsworth St., Granada Hills, CA 91344
Inside CA 818/360-3715 • Outside CA 800/522-2041 • FAX 818/360-1464



OLANIYAN

Color Separation and Professional Page V1.1

DESKTOP PUBLISHING ON THE AMIGA

In a market which is currently dominated by Apple's Macintosh™ the Amiga has had a rather difficult time competing with the little blue-screened machine. 'Why so?', you may ask, after all the Amiga has color, a multi-tasking operating system, graphics co-processors and several other features unavailable to the Mac user. The main advantage the Mac has had over our favorite computer is time, it's been around a few more years, this has allowed its software and hardware to grow and mature. The Mac has frame-buffers which allow full-page displays, it has several image scanners and dozens of Desktop Publishing packages. So what do we have? Well, color does make a difference, especially with the new release of high resolution color output devices. "How about the Mac II", you might say, "that has color and more power?". It does if you're willing to pay about \$12,000 dollars for it, anyway, you can expand an Amiga 2000 for less than that amount and still get a more powerful machine. Other advantages the Amiga has include easy expandability and enough graphic horsepower to leave the Mac standing in the dust.

As far as hardware is concerned there will be several new products available to aid the Amiga desktop publisher, for example, Commodore will

soon be releasing a monitor with its own frame-buffer; it is monochrome and will have a resolution of 1008x800 and will work on all Amigas. In terms of publishing software the only packages available have all been of an introductory nature. None have had the power and features needed to be of any adequate use in a professional environment. Enter Professional Page from Gold Disk Inc.

PROFESSIONAL PAGE V1.1

Finally! What we've all been waiting for, a way to turn our Amigas into fully functional and professional desktop publishing stations. To use Professional Page V1.1 we require at least 1MB of memory, and two floppy disk drives, a hard-drive would also be very useful to hold all the screen fonts, but it's not a necessity. The program works at a resolution of 640x400, 640x200, or can also open up on the Workbench screen (i.e. you can typeset at 1008x800 using the new A2024 Amiga monitor). The Professional Page screen will also open to overscan size, for example, here at AmigoTimes we run Pro Page at a resolution of 704x470 and on A2000 Amigas fitted with the MicroWay "Flicker Fixer" displaying onto NEC monitors, (NO FLICKER, WHATSOEVER). Pro Page comes on two disks, the first contains the program itself and the second contains 35 screen fonts, a tutorial, and a

By Eyo Sama

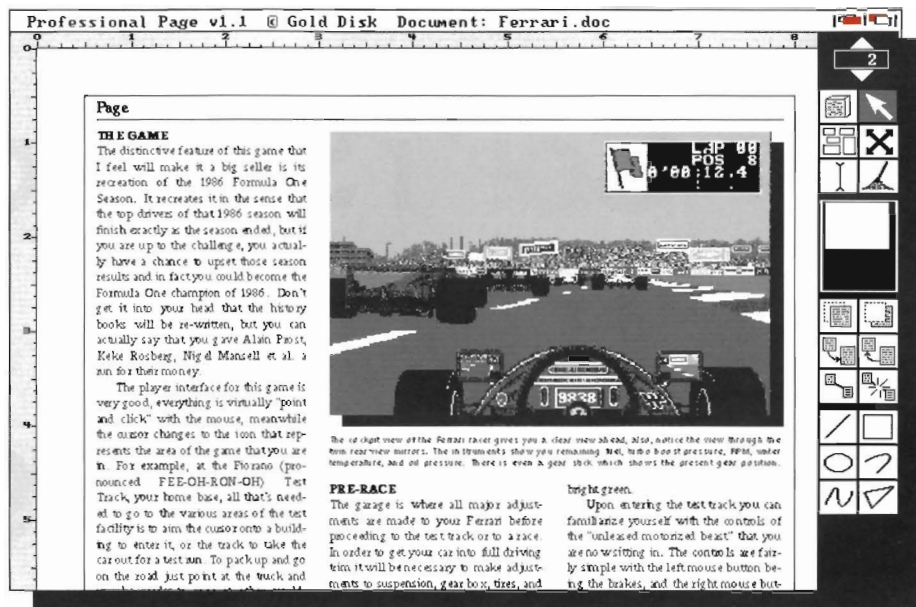
utility to produce different font sizes.

Professional Page V1.1 has several enhancements over V1.0, also several bugs which were found in the first version have been fixed. The best news is that Gold Disk has decided to incorporate their color separation module into V1.1, this means that color desktop publishing is now a reality for the Amiga and personal computers in general. It is now possible to print Amiga graphics in their full splendor and color; each image imported into Pro Page can be printed with its own palette and at the full resolution of the output device. To give you an idea of the capabilities of this program, flip through the pages of this magazine, it was entirely typeset using Professional Page V1.1; the artwork was produced using several Amiga paint programs and was then imported into Pro Page. The pages were then color separated (a mere menu selection) and printed directly onto film using a 2500 dpi (dot-per-inch) Linotronic L-300 type-setter. But don't worry, you do not need to own a Linotron or a Laser printer to see your Pro Page output because version 1.1 now supports dot-matrix output (even in color) and uses the new, improved, Workbench 1.3 printer drivers to do so.

When the program is loaded you are faced with a desktop and a strip of tools running down the right side of your screen. The first thing you have to do is create a page by selecting the appropriate menu item; the page then appears on the desktop with rulers along its top and left border. To actually place graphics and text onto the page you open up boxes into which you can load IFF pictures, Aegis DrawTM structured drawings, or text in several formats (Generic, no CR Generic, TextCraft, TextCraft Plus, WordPerfect, and Scribble). Documents are viewed in different resolutions and all editing, positioning, cropping, and word processing is done on the page in a WYSIWYG (what-you-see-is-what-you-get) environment.

THE PAGE

Pro Page allows you to create documents containing up to 999 pages (memory permitting), and each page can be between 1"x 1" and 17"x 17" in size. For



each page you can set default margins, number of columns, the size of the gutters between these columns, and the PostScript output specifications. The requester for the output specs allows you to set the scale, position, and rotation of the whole page. This requester will also allow you to set the size and position of crop marks should you wish to include them.

To aid the placement of boxes you can use a grid which is superimposed onto the page. The grid and the rulers, which can be made invisible, can be set to use inch, cm, or pica measurements. One very helpful feature is "Grid-Snap", which forces any drawing, box placement etc. to be aligned with the grid intersections. All this editing can be done at 5 levels of magnification: 200%, 100%, 50%, 33%, and 25%.

Very useful is the ability to set up odd and even page templates; you can create boxes holding columns of text, page numbers, logos etc. which you want appearing on every page of your document. These templates can then be saved to disk and will really save a lot of time when you're creating long documents containing similarly formatted pages. For example, the article you are now reading was created using such templates; the columns, the borders, and the page numbers were set up as boxes and saved as templates which we reuse for other articles. In fact, we have created and used a whole library of different templates to create this issue. It would

have been nice if Professional Page came with some pre-made templates.

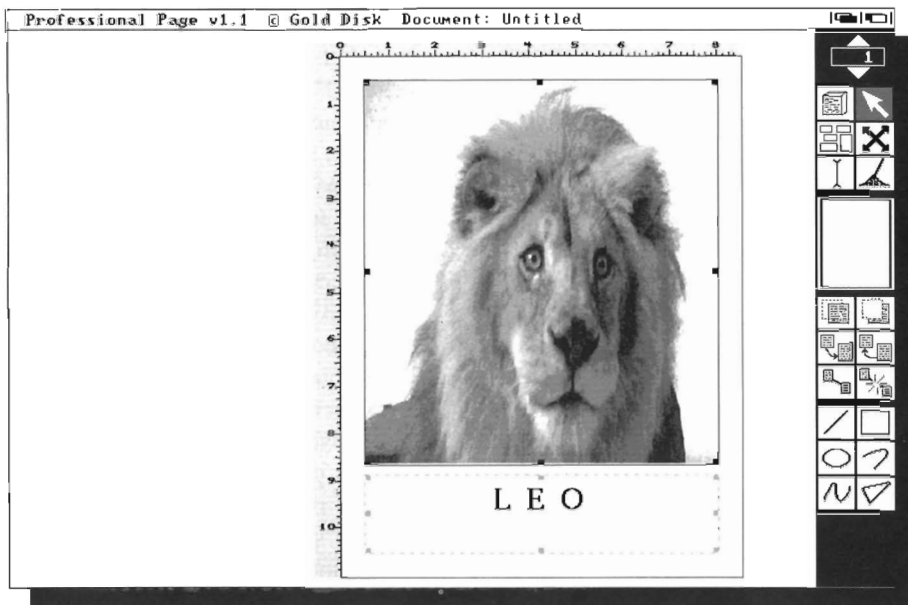
THE BOX

The "box" is the building block of Professional Page and is created by clicking the left mouse button where you would like the top lefthand corner to be and releasing it when you have the desired size. Once boxes are opened they can be accurately re-sized and cropped using handles on the corners and sides of the box. Using the box menu you can clone, delete, or alter the selected box. Double-clicking the left mouse button on any box on the page brings up a requester allowing you to alter the specifications of that box. The requester then allows you to numerically adjust the scale, the dimensions, the 15 tab positions, and whether text should flow around the box.

Several boxes may be grouped together, and thereby moved around and manipulated as one unit, (you can even merge them together as one box if they contain structured objects). Text within boxes can be made to flow from box-to-box and from page-to-page using the mouse to click on each box that you want linked.

WORD PROCESSING

All the text manipulation (word processing) is done on the page in whatever resolution has been set, what you see is exactly what you get. The only exception is text which has been written using



font sizes which are not present in the FONTS directory, the text is still in the right position and still uses the right spacing but the next available font size is used to display it. Type sizes up to 127 points can be handled, though most of the screen fonts supplied with the program are no larger than 48 points. For this, Gold Disk has included a utility, called MakeFont, which is used to generate smaller or larger point sizes from existing fonts. (this is a very useful utility and I have found myself using it for other purposes with other programs). Unlike its predecessor (Page-Setter), Professional Page lets you use different fonts, sizes, and colors within the same box.

The built-in word-processor has all

the usual editing features such as cut, copy, and paste. Text is highlighted using the mouse or by selecting the "select box" or "select all" menu items. This highlighted text can then be saved as an ASCII file. There is a fairly sophisticated search and replace function that allows you to find and replace text on the basis of color, font, font size, and style.

Each box can be set to automatically hyphenate the text it contains, and Professional Page 1.1 allows you to hyphenate in English, French, or Spanish. You can also create an exception dictionary containing words you do not wish to have hyphenated or words you would prefer hyphenated in a certain manner. The minimum prefix and the minimum suffix for a word to be hyphenated can

also be set.

As mentioned before Pro Page allows you to import from several other word-processors, you are in no way limited to just using the built-in word-processor. When importing text all the formatting (such as **Bold**, *Italic*, OUTLINE etc.) is retained. When writing on your favourite word-processor you can use special character codes to select font type, color, size and style, the kerning, hyphenation, and line spacing of the text. For example, "\c<RED>\B AmigoTimes \n" would print the word **AMIGOTIMES** in red and boldfaced. Of course, all these attributes can be selected from within Pro Page by highlighting the text and selecting the appropriate menu item.

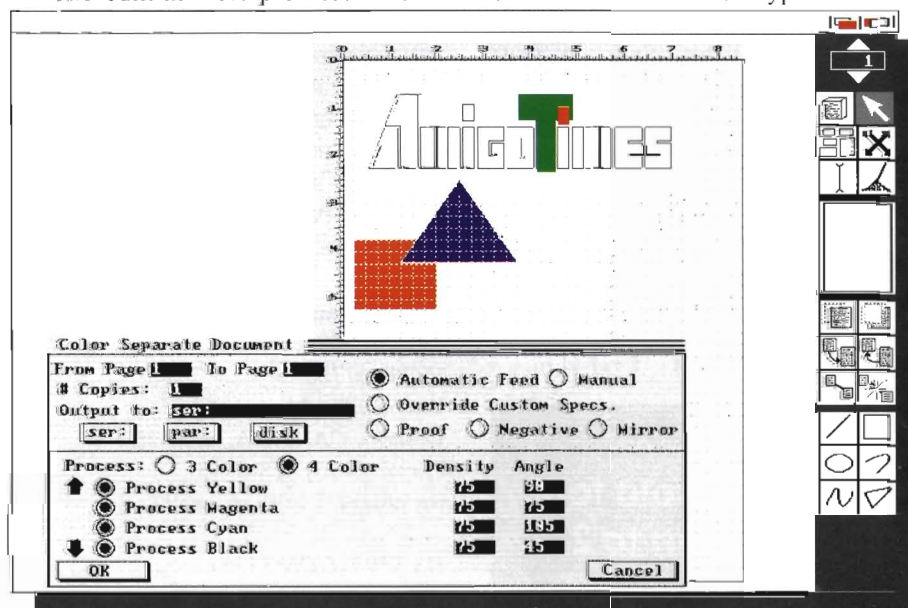
TYPESETTING

As far as typesetting features are concerned, Pro Page allows you to automatically kern a block of text, (kerning is the adjustment of inter-character spacing to make text more readable). The baseline can be shifted ^{up} or _{down} for individual characters or for whole blocks of text. The tracking can be adjusted by 1/100th of an em space (i.e. text can be *s p r e d* apart or brought *closer* together). Tracking and baseline shifting can be done numerically or by tapping on the cursor keys while the text is highlighted. Paragraphs can be indented or hanging, and they can be left-adjusted, right-adjusted, centered, or flushed.

GRAPHICS

There is an in-built set of structured drawing tools, these range from simple line, circle, and rectangle drawing, to a Bezier curve tool. The structured objects created can be made of various line weights, patterns, and colors. Up to 127 colors from a palette of 4096, can be defined and used from within Pro Page; this limit, of course, does not apply to BitMapped graphics, each of which can have its own palette of up to 4096 colors. Defined colors can be saved and then re-loaded for other documents.

If you're not satisfied with the included tools, you can import structured objects which have been created with Aegis Draw (or DrawPlus). Structured



objects, created either way, can be cropped and scaled using the box handles or via requester.

As far as BitMapped graphics are concerned, all IFF formats, including HAM, can be loaded, even resolutions up to 1024x1024 and 8 BitPlanes. On screen, graphics are displayed in 8 shades of grey, but the original color palette information is retained. As with structured graphics, boxes containing IFF pictures can be cropped and scaled numerically or by using the box handles. Text can be made to flow over or around boxes containing graphics.

COLOR SEPARATION

The Amiga has always been noted for its spectacular color displays, but how do you show-off those graphics without dragging around about 80 pounds of computer equipment. The built-in color separation module, is one of the most exciting features of this program. Once your page full of IFF pictures and text is ready, you need only make a menu selection to color separate your document. Line density and angles are options which can be set for each of the primary colors (i.e. Yellow, Magenta, Cyan, and Black). Each of the four colors is printed with accurate registration marks and an indication of what the color is. To get the best results though you need to print to a Linotronic typesetter; you have the option to print onto paper or film, and in negative or positive, and mirrored. As mentioned before, the pages of this magazine were all color separated using Professional Page 1.1.

PRINTING

With Professional Page 1.1 you can output to any PostScript device such as a Laser printer or Linotronic typesetter. You have the option to print the whole document, the current page, or even a thumbnail print, where you can get up to 16 shrunk pages onto one page. If you do not have direct access to any one of these devices you can always print your documents to disk files, which can then be printed, or sent by modem, to printing services which do. Graphics are printed as halftones and at the density and line angle set.

Another option is using dot-matrix printers. Professional Page 1.1 uses the new Workbench 1.3 printer drivers, these drivers are greatly improved, they print faster, more accurately and make better use of color and 24-pin printers. The dot-matrix output is very crisp, depending on the quality of your printer, and the color graphics look very impressive, especially on a Xerox 4020. A requester lets you set various printing preferences such as density, scale, dithering, and will even let you use anti-aliasing to smooth-out the "jaggies" associated with dot-matrix print.

WHAT FEATURES COULD BE ADDED

Now some enhancements I would like to see in future upgrades; the most evident limitation in Pro Page is the fact that text and graphics cannot be angled, you can sort of get the same effect by rotating and scaling whole pages and then superimposing them, but it is both cumbersome and difficult to do. Gold

Disk is aware of this, and I believe they will include such a feature in a future upgrade. The ability to view and edit two or more pages, on the screen, at the same time, would also be a useful addition, as would having a movable or hide-able tool bar, which would aid editing at higher resolutions.

Of course I have a small wish-list of features like having an in-built spell-checker and a thesaurus, but I just received GoldSpell 2.0 (also from Gold Disk) which allows you to spell-check Professional Page and PageSetter documents. Oh yes, I would also like to have automatic page numbering, and an alphabetical sort on the file requesters, it gets hard to find your files in directories containing 20 or more files, as is often the case on hard drives.

CONCLUSION

At last, with the help of Professional Page 1.1, the Amiga has the capability of being used as a serious color Desktop Publishing system. Not only is it the top Amiga Publishing program, but it can easily hold its own with top IBM and Macintosh packages.

The manual is well written, easy to understand, and was typeset using Pro Page. The version 1.1 addendum (actually a complete and bound supplement to the main manual) really beats the bunch of stapled sheets you get with some programs. With the upgrade to version 1.1, Gold Disk has shown that they are very serious about supporting and continuously improving this product. Owners of Professional Page version 1.0 can upgrade to version 1.1 at no extra charge; I find this a very commendable policy, because with the color separation and dot-matrix support, Professional Page 1.1 is an entirely new product, not just a bug fix. □

Professional Page version 1.1

GOLD DISK

P.O.Box 789

Streetsville, Mississauga,

Ontario, CANADA, L5M 2C2

Phone: (416) 828-0913

for orders: 1-800-387-8192.

US \$395 (CDN \$525)

1MB required, 2 floppy drives.

A B C D E F G H I J K L M

This example shows the Helvetica Bold font ranging in point size from 127 down to 2.



Have you ever wished for a paint program that has all the features and ease of use of Deluxe Paint II, with the added feature of working in 4096 colors? You know, something that has all the standard features of a paint program i.e. (flood fills, bezier curves, circles, polygons, custom brushes, etc.) but also does perspective manipulations, smoothing, blending, dither shading, object wraps around circles, tubes, cubes and irregular shapes. Yes, I know that I have just described most of the features of some \$20,000 Graphics packages, or even more expensive dedicated work stations, but those are just some of the features of Photon Paint from Microillusions.

PACKAGING

The overall presentation of the package is a definite plus, if those pictures on the box don't get your attention, nothing will; and keep in mind that with a little help from a digitizer and some of the features of this paint package you too

can create computer masterpieces like those pictured on the box. Inside the package are a manual and two disks; the Program Disk and a Sample Art and Tutorial Disk. The tutorial disk contains a series of sample drawings that are intended to be used with the tutorials in the manual, and some samples of artwork done using Photon Paint.

THE MANUAL

The manual is well laid out but has one major drawback, the noticeable absence of an index. The absence of an index led to some, at times, frustrating searches through the manual looking for a feature, or an explanation of how to use a particular function; a table of contents helps a bit, but its no substitute. On the plus side though, the Tutorials in the manual were very helpful and are strongly recommended. Apart from the tutorials giving practical hands-on experience, when you see the ease of use of some of the some of the features such as object wrapping, you are instilled with a certain amount of self-confidence and

By Olusegun Olaniyan

further explore the other features. The manual may at first seem tedious with its explanations of some of the most painfully obvious things, but keep in mind that some people are not as familiar with the Amiga as others; my suggestion is be patient and you may find that some of the manual's oversimplification is justified.

PHOTON PAINT

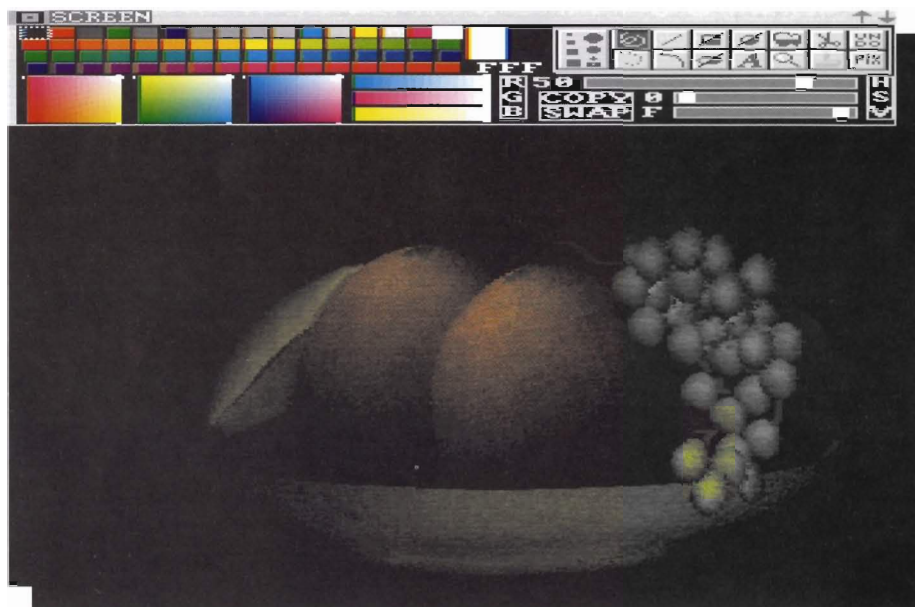
Quite a bit of the program is intuitive and to seasoned users of Deluxe Paint II it can be simplicity in itself, although I challenge you to figure out how to use the "Tilt" function without consulting the manual. The reason why I say that a user of Deluxe Paint II will almost feel at home is because the makers of Photon Paint kept an un-written standard of keyboard commands established by Deluxe Paint II. This keyboard standard will no doubt be greatly appreciated by artists that are familiar with Deluxe Paint because you don't have to keep memorizing new keyboard commands for every paint program that you use.

FEATURES

Photon Paint has a lot of standard features and some not so standard having some of the fastest pop-up requesters I have ever seen. You have a choice of four screen modes, 320x200, 352x240, 320x400, and 352x480. These modes all work in NTSC or PAL, (with PAL, of course, the resolutions are slightly higher). Photon Paint HAM files load, and save, very quickly.

Menu selection is done via the "fast menu", a handy menu that can be moved around the page or removed with a keystroke or a click of the mouse button. Some of the standard features are the "Free Draw", "Dotted Free Draw", "Line", "Arc", "Rectangle", "Ellipse", "Circle", "Flood Fill", and the "Scissors (Brush)", all that's missing is an "Airbrush" but some of the built-in features sort of make up for it. All features can be further enhanced in the "blend" drawing mode which enables you to do perspective fills while varying the amount of dither.

"Blend" allows you to blend the brush into the background thereby giv-



The Photon Paint screen displaying the Fast menu. The palette contains the 4096 colors that the Amiga is capable of displaying, all of which are accessible to the user with just the click of a mouse button. The intensities of the colors can be adjusted via RGB and HSV sliders.

ing it a transparent or ghost-like effect, this can also be done in conjunction with the perspective fill. When "Set" is selected in the mode menu you see a requester that allows you to set the perspective in either the X or the Y-direction or in both, and as a note the requester in this mode may seem alien to some and "really neat" to others, all I can say is experiment with it till you get the hang of it.

"Perspective" allows you to alter the orientation of a brush in 3-Dimensional space, if there is a difficult mode to master in the program, this is it; the whole operation gives the appearance of being complicated and nothing short of a "Top-Gun" fighter pilot will master it right away. As an example, to rotate a brush in perspective you must select "tilt", then lock the "pitch", rotate the brush to a desired angle (represented by an outlined box), unlock the "pitch" then lock the "roll", rotate the brush again to the desired angle and press the spacebar to draw the result; there's got to be an easier way.

One interesting feature that is not found on any paint program I know of is "Mouse Speed"; this allows you to vary the response speed of the mouse between slow, for tight work (i.e. magnify mode), medium, and fast in the normal work area. Another handy fea-

ture is "Follow Mouse"; when the mouse is being moved rapidly this ensures that the path of the mouse is traced exactly even if the line is not being drawn at the same rate as the movement of the mouse.

When drawing in overscan mode you can't see the full screen, but the "Screen Offset" menu selection allows you to adjust to what area of the screen you can view, this is not unlike the offset gadget in "Preferences" although I would have preferred using the scroll keys which are implemented in the magnify mode.

Now we get to one of my favorite features of the program, 3-D Surface Mapping, wrapping onto 3-D objects to the rest of us. This feature allows you to wrap a brush around one of the following 3-D objects: a tube, a cone, a ball, a cube, each separate side of a cube, an ellipse, or a symmetrical free-hand drawn shape. As an added bonus, you can adjust the intensity and direction of a light source shining on the shapes. When you select "Lum" in the menu, you can increase or decrease the intensity as well as change the direction of a light source including backlighting it to give it a glowing halo effect. Wrapping textured brushes around geometric shapes gives some very shocking effects, as well as some very realistic results;

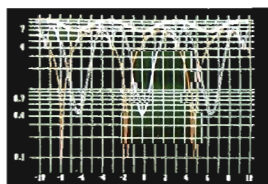


Descartes!

POWERFUL MATH RELATION GRAPHER!

Brings mathematical relationships, simple and complex, into vivid color. IFF compatible, so results can be used with paint programs and desktop publishing.

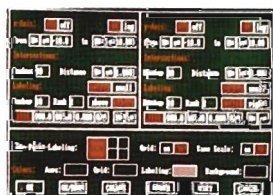
The program René would have written!



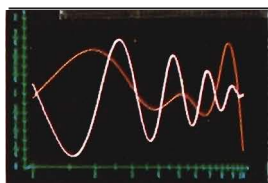
PARENTS...
enable your child to achieve greater understanding.

TEACHERS... an invaluable aid in lesson preparation and presentation.

STUDENTS...
have an instant visual aid to help grasp those new concepts



SCIENTISTS & ENGINEERS... functional relationships at your fingertips.



3D DESIGNERS & ANIMATORS
...gain deeper understanding of the math required to master the art

ONLY
\$34⁹⁵ U.S.
Sugg. List
\$49.95 Cdn. in Canada

MINDWARE INTERNATIONAL

110 Dunlop St., W., Box 22158,
Barrie, Ontario, Canada L4M 5R3
(705) 737-5998

Also from Mindware — Animate your full-screen IFF images with **PageFlipper™**

Available from your Amiga Dealer.

Mindware Products distributed by:

American Software Distributors, Urbana, IL 1-800-225-7941
— in Illinois (217) 643-2050



Holmes!

THE ULTIMATE AMIGA DISK UTILITY!

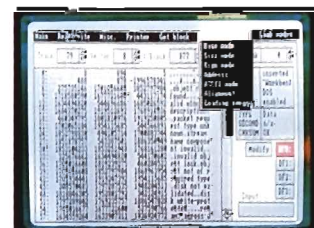
You will wonder how you ever did without it! Discover the mysteries of your Amiga's file and disk structures.

Sherlock wouldn't need Watson, if he had this program!

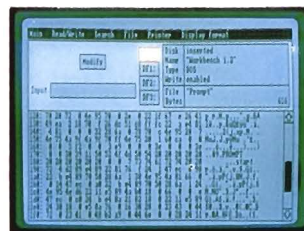
File	Size	Free	Used	Free	Used
Root	1024	1024	1024	1024	1024
...

DISK & FILE REPAIR... lets YOU control the recovery of a bad disk or lost data.

DISK & FILE ALTERATION...
change messages in programs — easily modify data.



DISK & FILE EXAMINATION
... program, data files and disk structures — all at your fingertips.



ONLY
\$49⁹⁵ U.S.
Sugg. List
\$69.95 Cdn. in Canada

CSS East, Cambridge, MA — 1-800-343-7535,
in Mass. (617) 876-2505

CSS, Chicago, IL — 1-800-422-4912, in Illinois, 1-800-331-SOFT
Southern Technologies Inc., Dallas, TX — (214) 247-7373

Amiga is a registered trademark of Commodore Business Machines Inc.

for example I stayed up late one night creating a very realistic looking sunkist orange and apple (ask any staff member).

One last feature that I cannot let pass uncomplimented is Photon Paint's ability to re-assign the font library to any disk other than the System disk, without having to quit the program, re-assign the fonts library, and reload; this is a feature that I hope other parties, whom shall remain nameless at this time, will adopt.

CONCLUSIONS

At this point after playing around with Photon Paint, I can honestly say that it is a definite "must see". Photon Paint is not the first HAM paint program, but it is one of the more impressive ones, fairly easy to use with features that are known to users of other paint programs. Since the release of Photon Paint there have been rumours of other HAM paint programs including believe it or not Photon Paint II, and DigiPaint II. In an-

swer to these rumours, all I can say is until we see anything this will be my choice for a HAM paint program.

At this point in time, because of Photon Paint's similarity to Deluxe Paint II, a comparison will almost certainly be made, but one thing that must be kept in mind is the fact that Photon Paint is as stated before, a HAM paint program whereas Deluxe Paint II is not, each program had different constraints and guidelines when they were being written and therefore they can't really be compared. Knowing this, Photon Paint cannot be seen as a replacement for Deluxe Paint II nor is it, but rather they complement each other quite well. For example you can do your initial work on Deluxe Paint II, which is still the king of PC paint programs, and then pass that work to Photon Paint to give it an added dimension in the HAM mode. Photon Paint with its features is definitely a program to be contended with, it offers a lot of the features of higher priced graphic workstations at a rea-

sonable price on a reasonably priced computer. I'm not saying that you should get rid of your high priced Graphic Workstation to opt for an Amiga bundled with Photon Paint, not by any means, but rather those companies that are selling high-priced Graphic Workstations or Graphic Packages should now have something to seriously think about regarding their pricing policy. □

Photon Paint

Microillusions, Inc.

17408 Chatsworth St.

Granada Hills, CA91344

Inside CA (818) 360-3715

Outside CA (800) 522-2041

\$99.95

512K required

(1MB or better is recommended)

JUPITER TECHNOLOGIES inc.

- * Friendly Services!
- * Ongoing Support!
- * Aggressive Pricing!
- * Always on the Top With . . .

AMIGA 500



**ON SITE
REPAIRS!**

AMIGA 1000



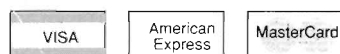
*** LARGEST SOFTWARE HOUSE *
ACCESSORIES GALORES
HOT NEW PRODUCTS**

**KNOWLEDGEABLE
INFORMATION!**

AMIGA 2000



**EDUCATION
DEALERS!**



**1-514-695-9124
1-514-695-9123**

**COMPLEXE POINTE CLAIRE
6321 Trans Canada Hwy.
UNIT #132, Pointe Claire
Que. H9R 5A5, CANADA**

MIDI

Computers have revolutionized the world of music. Since the invention of synthesizers people have been producing sounds and music with the help of electronics, and now with the microcomputer, the possibilities are endless. More and more, computers are being used in every aspect of music making, from controlling a single instrument, to the control of an entire studio or stage setup. Now with the acceptance of MIDI (Musical Instrument Digital Interface) as a standard, there is no turning back. One important benefit of this revolution is that it is now feasible and affordable for the amateur to build himself a professional quality mini-studio.

Because the computer will be the controller of the MIDI based music system, it is important to choose one that's as versatile as possible. Of course, the Amiga is a good choice, not only for its good sound quality, but also because of its multitasking capability, which makes it possible to run different applications at the same time, as long as there is enough memory. I think multitasking will become an important factor in future hardware and software development; proof of this is that every major manufacturer is developing multitasking operating systems for their machines. Software developers will now have to readjust their designing philosophy and think in terms of modules exchanging their data rather than the "do everything in one program" approach.

There are many ways in which a computer can be helpful in the making

of music, either by itself or as a controller of devices such as synthesizers, sound modules, drum machines, and even mixers, all with the aid of the MIDI standard. By itself the Amiga can do many of these things, you can write your music on it and use the Amiga's stereo output to listen to your masterpiece. But you don't have to stop there, you can widen your musical horizons by adding a MIDI interface, and either a MIDI synthesizer or sound module. In some cases, you'll need other programs to make the most of these additions.

One of the better known applications is the sequencer. A sequencer is used to record and store musical data through MIDI. Sounds are not recorded, but all the actions on the keyboard to produce them are recorded as MIDI messages, either in real time or in step time (one note at a time). Using a sequencer you can edit what you've recorded and you can also record tracks one at a time so that in the end it sounds as if many musicians were playing simultaneously. There are several sequencers available for the Amiga, so you'll have to look carefully before buying.

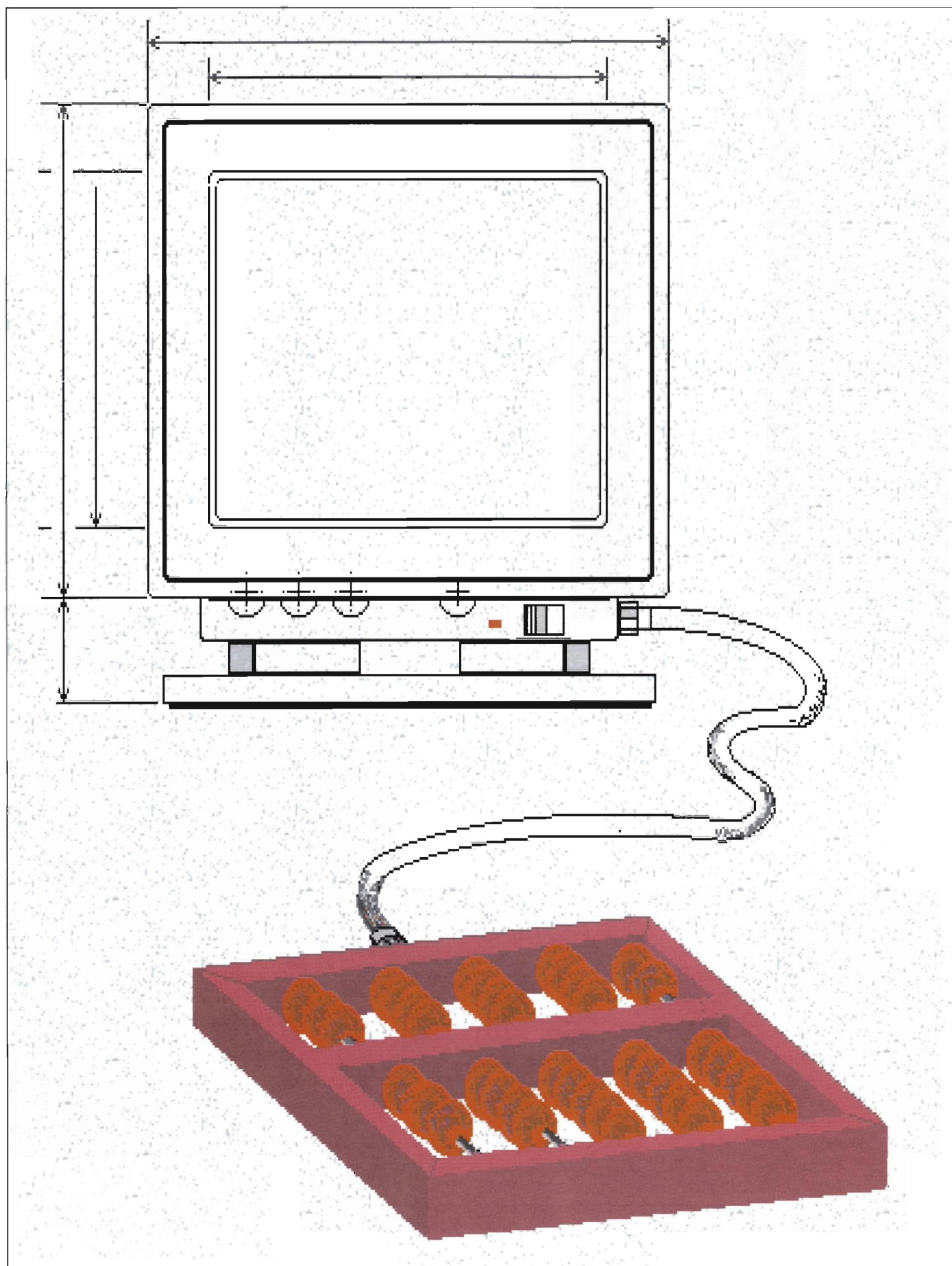
Synthesizers usually contain many preprogrammed sounds which are difficult to program directly. In this case you can use a patch librarian. A patch, in synthesizer talk, is a particular set of settings that allow a specific sound to be produced. With a patch librarian you can use your computer to transmit preprogrammed patches to the synthesizer. The patches are then transmitted from the computer to the synthesizer through

MIDI. However, because most synthesizers program sounds differently, and the patch data is sent using system exclusive data, a patch librarian is specific to a certain synthesizer. Unless the librarian itself is programmable, you will need a different librarian for each synthesizer you own.

Another way to use your computer with MIDI is to program the synthesizer with the computer rather than doing it directly on the synthesizer or with a dedicated programmer. The necessary program should be able to vary the different parameters in a patch and then send this data to the synthesizer. Here you also need a different programmer for each synthesizer. Very often a program will contain both the librarian and the patch programmer. Unfortunately there aren't many of these applications available for the Amiga. Some are becoming available, but they are ports from other computers.

These are just a few of the ways you can use a computer with the MIDI system. As I said before, the Amiga is a very good choice as a music computer, it has a lot of potential, but the software is slow in coming. In future articles we will be looking at ways to program the Amiga for use with the MIDI standard, starting with simple examples and always keeping in mind that the Amiga is a multitasking machine, and that a particular program is only a building block of a complete system. I will also keep you informed of the programs becoming available for the Amiga, and we will also take a close look at MIDI hardware that you can add to your system. □

By Serge Boucher



THE HISTORY OF COMPUTERS

"A history is by its nature an approximation to the reality that was." (N. Metropolis and J. Worlton)

I believe it is important to provide Amiga computer owners (and other interested microcomputer owners) with "The Big Picture". The spearhead of this article is aimed towards the theories that helped mold computer science, and how, and by whom some of the technology was developed. It's a well known fact that throughout history those with access to knowledge have attained greater heights of power and wealth. Combine this with the fact that computers have the capability to access any type of information much faster than any other form of information gathering system, and what you have is an important tool for success. This continuously expanding market of personal computer owners has had a large impact on telecommunications and soon it will cause another explosion in that field. These explosions are creating a greater demand for better, faster, smaller and more powerful computers and this in turn demands improvements in telecommunications.

THE DEVELOPMENT OF OUR NUMBER SYSTEM

How did all this come about? It definitely didn't happen overnight! Many important theories, developments and inventions preceded the dawn of anything that we call computers today. Even before any form of mechanical computation devices were invented, man first had to invent some form of number manipulation system. It was the lack of currency in early centuries that led to this need. Back then, civilized societies usually used detailed accounting systems to pay their workers in grain, beer or other forms of essentials, and also to perform other business tasks. Only highly developed societies would have a need for and the impetus to develop a number manipulation system.

It should come as no surprise that the Chinese developed such a system, when you consider that they invented gun powder, the plow, produced silk, built the "Great Wall" and provided some of the greatest philosophers in history. By 3500 B.C., the Chinese had de-

By Ernest Nagy

veloped sophisticated mathematics but it was only around 600 B.C. that any major developments occurred. By this time, they had a well-developed decimal value system, and used positive and negative numbers. Some time later, they borrowed the number zero from the Arabs.

In yet another part of the world, the Egyptians developed their own version of a number manipulation system, based on 10. It was difficult to write numbers in hieroglyphics quickly so the Hieratic script (a shortened version of the hieroglyphic script) was used for mathematical manipulations. The "Great Pyramids of Cheops" were most likely built using this math. Other great civilizations, including the Greeks and Romans, had their own versions of numeric manipulation systems. Even though these civilizations were able to achieve mathematical feats that engineers marvel at to this day, their numerical systems lacked simplicity of use. Today, mathematics is the only universal language understood by every civilized nation.

The exact date when humans began to use the current number system is still unclear, but it is known to have arrived in Europe in 1250 A.D., from the Arabs. They in turn obtained it from the people on the Indian subcontinent - where the Indians obtained it is still a mystery.

FIRST CALCULATING TOOLS

The development of numbers was great but man still had to manipulate them in his brain to achieve the final result. Sure there were some errors and yes, it was slow, but what else were they to use, their fingers? Absolutely! Because the Chinese were the first civilization to have some form of numerical system, it was only natural that they should develop the first calculating tool (called calculating rods). It was probably invented by the poor soul whose job it was to calculate the number of bricks to be used in the building of the Great Wall. Calculating rods, known to have existed since 400 B.C., were a collection of 256 rods - could that have anything to do with there being eight bits in a byte? The rods were colored either red or black: the red stood for positive numbers and the black for negative num-

bers. They were fashioned of either bone or ivory and were manipulated in some manner to produce mathematical results.

The abacus, also invented by the Chinese, was the next step forward and it eventually replaced the calculating rods around 1200 A.D. The abacus came in many shapes and varieties, but its most popular form consisted of a frame holding parallel rods strung with moveable counters (beads were used). The abacus eventually became known to the Romans and early European traders, initially adopted by westerners in a different form, in which stones (called *Calculi* in Latin, hence the word *calculate*) are moved around in grooves on a flat board. The abacus is still the most widely known calculating instrument; in fact it is still in wide spread use throughout China and the USSR.

A WHILE LATER...

The next major step in calculating devices was the invention of a tool called a "sector". The famous astronomer, Galileo Galilei, made the most useful and widely copied version of the early sectors. He started work on it in 1597 and before 1610 had manufactured over 300 pieces for commercial sale.

In 1614, John Napier (a Scottish mathematician) invented logarithms and introduced the decimal point in writing numbers. In 1617, Napier introduced a mechanical device called Napier's Bones. The instrument had nine square sectioned rods each divided into segments. The rods were numbered in arithmetic sequence thus enabling multiplication by adding together the proper figures shown on the segments. This became the forerunner of the slide ruler which was invented in 1630 by William Oughtred, and at the time it was circular and still a bit crude compared to what some of us remember a slide ruler to be.

The first attempt at designing a mechanical analog digital calculator was attempted by Wilhelm Schickard (the German astronomer and close friend of Johannes Kepler). In late 1623, Schickard designed and built a mechanism that could add, subtract, multiply and divide. This device was never commercially available but was none the

less used by himself and some of his close friends.

Between the years 1642 and 1644, Blaise Pascal (a French mathematician and philosopher) invented and built a similar device to that of Schickard's, he named it "Pascaline". It was quite a large improvement over Schickard's device; it could handle addition and subtraction of numbers up to 999,999.999 and was about the size of a shoe box. Pascal had created the Pascaline to help his father with tax computations.

COMPUTER CONCEPTS EMERGE

The earliest known reference, in the English language, to the word "computer" was coined by Sir Thomas Browne back in 1646. It was used as a reference to people who recorded the passage of time by drawing up calendars.

Up to this point all calculating devices invented worked on the digit principle, but this was all to change due to a theory that was conjured up in the mind of Gottfried Wilhelm, Baron von Leibniz. Baron von Leibniz was a mathematician and diplomat, who was a rival of Isaac Newton, due to the fact that he independently developed calculus around the same time period. Upon his death in 1716, Baron von Leibniz left some important notes and ideas regarding binary mathematics. He reasoned that everything in the universe either did or did not exist, therefore existence could be reduced through binary arithmetic. The digit 1 meant existence and 0 meant non-existence. Unfortunately, Baron von Leibniz never succeeded in fully working out this math, but he had planted the seed for others to cultivate in their minds.

THE INDUSTRIAL REVOLUTION

By now the industrial revolution was on its way and manufacturing had to be sped up. The idea of using punched cards to control a machine's process was first used to control patterns woven by automatic looms. In 1728, Falconin linked a series of cards together (known as Falcon's Cards), each contained the code to automate the production of some pattern on the loom. Falcon's cards were successfully implemented by Joseph Marie Jacquard (a French silk

weaver) to control the first fully automated loom in 1801.

In 1822, Charles Babbage (the "father" of the computer) completed the design of his first machine "The Difference Engine" which was used in the computation of tables. A year later, Babbage began to work on the first modern computer called "The Analytical Engine". Even though Babbage worked on his computer for 48 years (up until his death), he never completed it. However, from all of Babbage's work came some important principles, who's major components were:

- 1) *The mill,*
- 2) *The store.*

The mill acted in much the same way as a CPU (Central Processing Unit) does today. It performed the same tasks as today's arithmetic unit and sequencer controller unit (by the use of conditional logic or branching logic statements). The store acted in much the same way as data memory does today. The mill would work on something, come up with a solution, pass it to the store and when it needed that solution, it could re-access and display any of its 1000 stored fifty digit numbers. The analytical engine was to derive its power from a steam engine and it used punched cards. The main principles of the analytical engine are the same elementary units used in serial processors today.

If you consider the importance of Babbage's work at the time, you might think that he would have left detailed notes. On the contrary, his notes were vague and his drawings unorganized. The world is indebted to one of Babbage's friends, Ada Augusta the Countess of Lovelace (The computer language ADA is named after her), for providing notes on Babbage's work. Apparently, Babbage gave a lecture in Italy on the analytical engine and L.F. Menabrea (a military engineer) wrote down the outline of its operation. This was eventually published in a Swiss periodical, which was later translated by Ada Augusta. Not only did she translate it, but she added an extensive set of comments and notes clarifying the detailed workings of the computer. It was

these notes which were later published as "Observations on Mr. Babbage's Analytical Engine".

During the time Babbage was working on his computer ideas, two other important developments occurred. The modern day slide ruler was perfected and boolean algebra was developed. In 1850, Amedu Mannheim perfected the slide ruler. The task was actually rather simple and intuitive - make the calibrated piece small and then add a movable cursor to it that would slide over the rectangular part. This made it look much like the slide rulers we know today. In 1854, George Boole (an english mathematician) fully devised binary algebra's principles, it had taken about 138 years to pick up where Baron von Leibniz left off with binary math.

Then came the first fully functioning computing device invented by Herman Hollerith (an american inventor) in 1890. Hollerith built a tabulating machine the size of a small office desk (for the U.S. Census Bureau) that added up census figures automatically by sorting and collating punched cards according to the positions of the holes on the card (this is the first known use of punched cards for data processing). The machine used electricity to drive the punched cards along. It took his machine two years to do what usually took ten years the old way. In 1896, Hollerith formed a company (Tabulating Machine Company) to build and market his invention and so started the automation of offices of America's biggest companies. In 1911, the company became "Computing Tabulating Recording Company". A series of mergers transformed his company into the famed IBM corporation (International Business Machines) in 1924.

It was William Thompson, Lord Kelvin (a british mathematician and physicist), in 1876, who was the first person to attempt building an automated machine that would solve differential equations mechanically. His machine was called a "Differential Analyzer" and was built from cogs, sprockets, rods and other such mechanical parts. The machine had a series of integrators, each one connected to the next one and there each integrator would contribute a small part to the final answer. Every-

thing worked except for "the integrator" - the main and crucial part.

William Seward Burroughs, in 1892, patented the first commercially successful adding machine. It included a full keyboard and a printing device to record the number totals.

About thirty-five years after Lord Kelvin's innovative idea first saw light, it was finally going to be finished by Vannevas Bush (a New England engineer at MIT) and his associate Norbert Wiener (a mathematician). Both Bush and Wiener were colorful people, but Wiener's background is far more interesting. Norbert Wiener learned the alphabet at eighteen months and by the age of six he was reading Charles Darwin (a formidable task for any high school student). By age eleven, Wiener was enrolled at Tufts University, and at the age of eighteen he had a Ph.D. from Harvard and a double degree in mathematics and philosophy. Later on when he became a professor, his students called him the absent minded professor.

In 1930, Bush and Wiener built the machine that Kelvin had envisioned. They attached a servomotor to the integrators main rod. This was done to provide the necessary torque that Kelvin never would have been able to produce. It worked marvelously except that every time a problem got larger more integrators had to be added. Of course, time had to be spent before each session to adjust, lubricate and align all the mechanical parts.

Wiener saw similarities between the brain and the workings of the computer and called his idea "Cybernetics" (from the greek word Kybernetes, meaning "steersman") and defined it as "the science of control and communication in the animal and the machine".

PART TWO

Next issue, the final installment of this two-part article, beginning with the second World War's role in helping to usher in the present day computers, and concluding with the emergence of microcomputers. □



IntroCAD

If you've owned your Amiga for any period of time you probably already own a paint program. In fact you might own many paint programs. But chances are you don't as yet own a drawing program. Up to now you might have not had any need for one, or been turned off from buying one due to the limited selection available and high prices.

IntroCAD is a new "object-oriented drawing program" from Progressive Peripherals and Software which is packed with features and priced reasonably enough so that any Amiga owner with a need to create line drawings of high precision and detail can now afford to. IntroCAD fills a vacant gap in the repertoire of available Amiga software and for its price is a welcome addition.

Shipped on one non-copy protected disk with a rather skimpy but well-written 50 page "lay-flat" manual, IntroCAD will mount easily on your hard drive, if you have one, and does not require any special configuration files to be put in any particular directories.

IntroCAD at first glance appears to be a very simple program - almost deceptively so. After working with IntroCAD for a while, you'll most likely agree that aside from some of the more esoteric features found in the expensive

CAD programs, IntroCAD is quite capable of getting your job done easily, with an elegant simplicity of style.

CAD VS. PAINTING

Unlike a paint program, in which you manipulate pixels of different colors on a screen, and then save or print the entire screen, an object-oriented drawing program like IntroCAD lets you draw lines or shapes which are defined not in terms of colored pixels, but in terms of "this line starts here and ends there". Since the output from a paint program to a printer is basically a dot-for-dot reproduction of what is seen on the screen, resolution is extremely limited even by the best printers, and even in the highest modes of screen resolution. What looks good on a paint program's screen rarely transfers well to printed output. A pixel has a finite size.

However, an object-oriented program is not limited to the resolution of the screen on which you are drawing. The screen simply serves as an interface so you can see what you are drawing. IntroCAD can use the highest and most detailed resolution of any output device it knows how to talk to, simply because the "picture" you've created is not stored as a "screen" but as a compressed list of coordinates for all the lines, ob-

jects, and colors you've used to create your drawing.

One other important difference between a paint program and a drawing or CAD program is that in a paint program, when you have created an object of any complexity, and then draw another area on top of it, the underlying object is overwritten by the new object. For example: draw a box, then draw a circle overlapping part of the box. Now try to move the box away from the circle. It can't be done in a paint program because the circle has overwritten the part of the screen where the box was. You can grab the two as a brush and move both of them, but the pieces of a picture in a paint program lose their individual identity when you start stacking things on top of each other.

In a CAD program, each object retains its own identity. No matter how complex your drawing becomes, you still have the ability at any time to grab an object, even with many other objects sitting on top of it or partially or completely surrounding or covering it, and move that first object to another part of the screen, or duplicate it, resize, recolor, erase, or manipulate it in many other ways.

IntroCAD operates in Amiga's "high resolution interlace" mode on a

By Harv Laser

screen of 640 pixels by 400 pixels. Screen flicker is minimized due to the built-in neutral background color (which you may modify), yet the screen never appears too dim for long stretches of work. Screen refresh is, depending on how large your drawing becomes (how many objects, lines, etc.), much quicker than in any other Amiga CAD program I've seen.

Like any graphics software, IntroCAD can use up a lot of RAM very quickly, again depending on the complexity of your drawing. While it will work with 512K, you should be aware that you might find yourself limited in how much you can accomplish until you add additional memory to your Amiga. Long ago I upgraded my Amiga to 2.5 megabytes of RAM and it made a world of difference not only in my ability to use the more sophisticated features of many graphics-oriented programs, but also in the entire "feel" of how my Amiga operates.

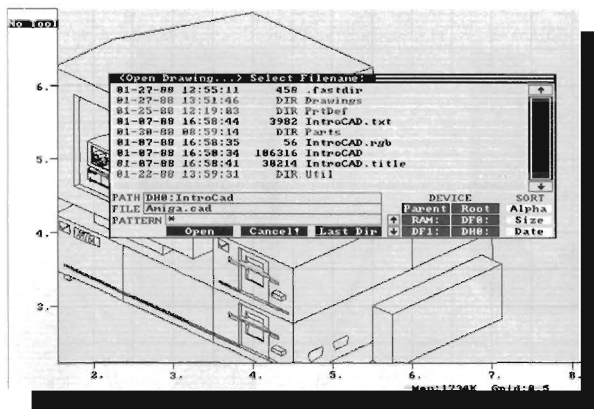
TOOLS GALORE

IntroCAD comes with a full arsenal of tools for creating and manipulating objects. All of the basic "primitives" are a menu-click away: lines, boxes, circles, arcs, free-hand drawing, and text. Each drawing tool is active until you choose a different one, and a screen prompt always tells you which tool you're using. A nice touch.

Once created, objects can be manipulated by such commands as erase, move, clone, size, rotate, color, set line type, and so on. Many types of grids are available as a backdrop along with two methods of "grid snap": snap to grid, or snap to drawing. Snapping in a CAD program means to tell the program to keep lines and objects aligned with the grid or with each other, instead of trying to manually align everything yourself.

For instance, try to draw many parallel lines by hand and keep the distance between them uniform - nearly impossible without the snapping feature. IntroCAD allows you to turn snapping on or off at any time.

Besides the menu selections for creating and dealing with objects, IntroCAD provides many tools and choices



The Incredible Pathmaster file requester, written by Justin McCormick, is used in IntroCAD. It automatically adjusts to any devices you may have attached to your file-system, and it will sort your files with respect to file size or alphabetic order.

to let you modify objects or groups of objects together, or to deal in different ways on your entire drawing at once.

A wonderful "Zoom" tool lets you zoom down and back out of any area for performing incredibly detailed work. (See the accompanying screen shot for an example of zooming into a drawing) "Group" lets you save one or more objects together as a "part" to be saved to disk (for use in other drawings) or to be dragged about the screen, duplicated, rotated, cloned, resized, or modified in many other ways.

IntroCAD's "Group" tool is ingenious. Once an object or multiple objects (and remember, objects can be anything on the screen including text) are grouped, IntroCAD will remember them as a group, unless you ungroup them.

Suppose you are creating a logo and have drawn a star with a circle around it and have grouped the two together. You now wish to resize this group and duplicate it elsewhere in your drawing. By clicking your mouse on the object, you will be presented with a new mouse pointer that looks like a "Y/N?" question. IntroCAD is asking you if you

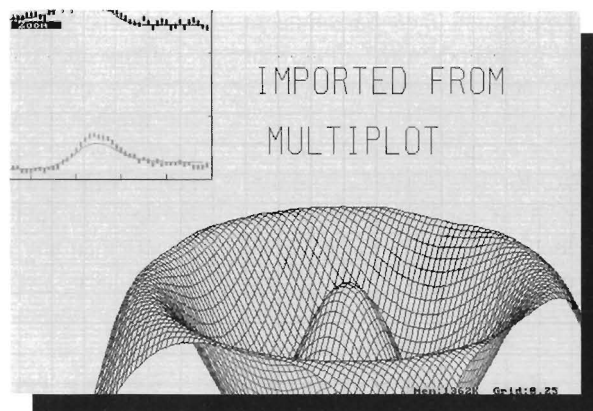
mean "this object here, or that one next to it" and will highlight each one while you click your left mouse button to mean "yes, that's the one I want" or your right mouse button to mean "no, show me the next one". This feature is fun and easy to use and no matter how complex your drawing becomes, you can always easily find exactly the object or group of objects you want to manipulate.

Any object can be colored through the use of a color requester, and colors may be modified with a palette requester. Lines may be solid or various patterns of broken lines can be used to distinguish objects from each other.

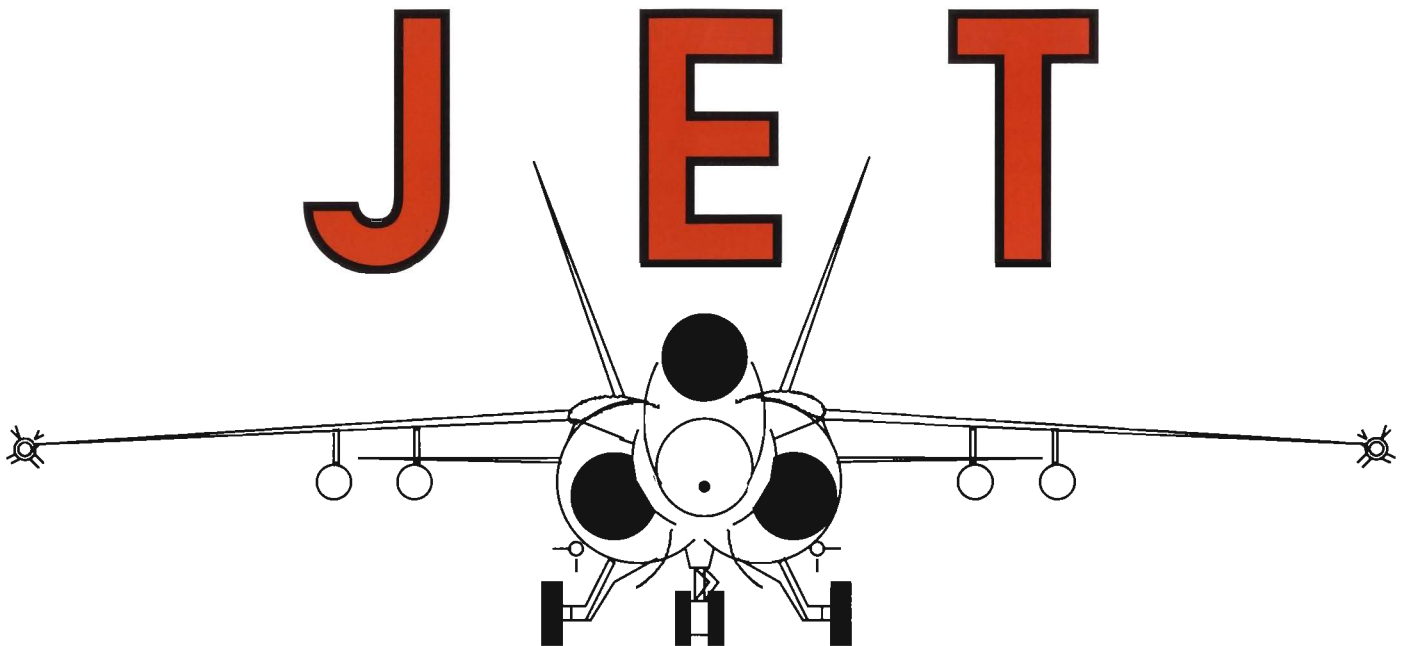
THE INCREDIBLE PATHMASTER

IntroCAD does not save drawings in the standard ILBM-IFF picture format used by paint programs, nor can you load, say, a Deluxe Paint picture into IntroCAD. This is not really a handicap as explained earlier, since IntroCAD stores its files in a format to deliver to you the best possible printer output, which is not a direct dump of the entire screen.

(continued on page 60)



IntroCAD comes bundled with two freely distributable plot generating programs. Output from these programs can be imported into IntroCAD.



After a long delay (nearly three years), SubLOGIC has finally released an Amiga version of their popular game, Jet. This three-dimensional aerial "shoot 'em up" with stereo sound, seats you in the cockpit of either a land-based F-16 Fighting Falcon or a carrier-based F-18 Hornet. You can dogfight against multiple Soviet Mig-21 and Mig-23 jet fighters, or attack land targets with the F-16 and Soviet cruisers with the F-18. All of the enemy aircraft and ships are detailed and three-dimensional. Some aspects of the game, such as mobile control towers, spot plane, multi-player option, etc., are similar to features available in SubLOGIC's other aviation game, Flight Simulator II. Even an ejection seat was included. If your plane gets shot down you'll have seconds to eject or you'll die. Your pilot will then parachute to safety. The game ends when all of your planes have been destroyed and you're awarded medals.

Once you've taken a quick look in the manual for the keyboard, mouse, joystick controls, and also for enemy and weapon information, you'll be set to go. For more realistic control and game play, I suggest you use your joy-

stick as the yoke instead of the mouse. Jet is not an easy game to play. When you first start the game, it might be best to start at the practice level (the enemies don't fire and you can't crash) in order to become accustomed to the super-sensitive plane controls. Once you've tried that, select the relatively easy level 1 - this time they bring weapons. If you want to get a better understanding of how to dodge enemy missiles, you could try watching the action from a control tower that you place a few thousand feet above your combat area.

The game offers nine scenarios and ten skill levels. Each plane has a scenario for dogfighting against Soviet Mig-21 and Mig-23 jet fighters, a target strike scenario (enemy factories or Soviet Kynda-class missile cruisers), and a free flight scenario for the pure joy of supersonic flight.

Only the F-16 has a "combined-attack" scenario that combines a bombing mission - watch out for the SAMs (surface-to-air missiles) - in addition to aerial combat with MIGs. This particular scenario contains more scenery and targets than any other combat scenario. You'll see pyramids, rivers, distant mountains, a sea, bridges, factories, two

By Michael Hermann

airports, and even a nuclear power plant. However, all this scenery occasionally slows the display and it might cause you to over-compensate some of your actions, which can be annoying or even fatal.

Another scenario accepts any Scenery Disk (see review on p.20) or the Flight Simulator II disk itself, however, there's no combat mode available while using these disks.

The navigation instrumentation is practically non-existent, all you're given is a compass. SubLOGIC neglected to include a navigation computer or, at least, a true ADF (Automatic Direction Finder) radio. While this isn't significant in the combat scenarios, it can be annoying when you're flying at night with a Scenery Disk.

The multi-player scenario is a one-on-one dogfight against another player over the modem lines or, if you happen to have two computers, by a direct cable hook-up. If you like, you could practice formation flying with the other player. But the scenery is plain land and only the F-16 fighters can be used.

THE BUSINESS END

Any modern day jet fighter pilot knows that speed can save his life. He finds his target, fires the missiles, and leaves immediately! It can also kill him. In Jet, as in real life, if you turn your plane too quickly at high speeds you'll feel the Gs (indicated by either a black or red flash on the screen). Not only is this increased weight uncomfortable, but a sharp turn can make your blood so heavy, that it will stop flowing - causing you to temporarily lose consciousness. Until you recover, you'll be easy prey for any missile coming your way or you may crash, so be careful!

Both planes come equipped with just about everything needed to trigger an international incident. In addition to a Vul-



can M61 20 mm machine gun (500 rounds), either jet can hold up to five of each of the four weapons best suited to your mission. You shouldn't be too quick to arm yourself to the teeth because the increased weight will decrease your maneuverability. Two pieces of equipment noticeably absent are chafe and flares. Chafe are strips of metal foil that are released to confuse radar-guided missiles and flares are used to throw off heat-seekers. Instead, you must continually rely on sharp turns and fast climbs to shake off an enemy missile. As a result, in the higher levels, it's much more difficult to aim at ground targets with all that movement.

GROUNDING FOR REPAIRS

As good as Jet is, a few things disappointed me. The game appears unfinished and not too flexible. For one thing, you can't choose your spot plane position or playback the last minute or so, as in Flight Simulator II. In the multi-player scenario, it might have

been more exciting if it was possible to use a Scenery Disk or the F-18 Hornets (at sea with the carrier). In the cockpit, it stops short of the level of realism found in Flight Simulator II. Most of the HUD (Heads Up Display) is accurate, but the border for the "instrumentation" is totally wrong. I would have appreciated some more relevant information on the HUD and a separate, realistic instrumenta-

tion panel.

More importantly, I came across too many bugs. For example, while the plane is on the ground, the spot plane won't properly show your plane's position. The jet won't go any higher than 50,000 feet even if the plane is tilted straight up. The enemy planes never crash, even if they dive into the ground (now is that fair?). In both of the target strike scenarios you can't use the two weapons (AGM-65 Maverick air-to-ground missile and the MK-82 Smart bomb) that you're supposed to use because they almost always take a detour and head north. If you contact SubLOGIC you can get an upgrade to 1.01 which fixes these bugs.

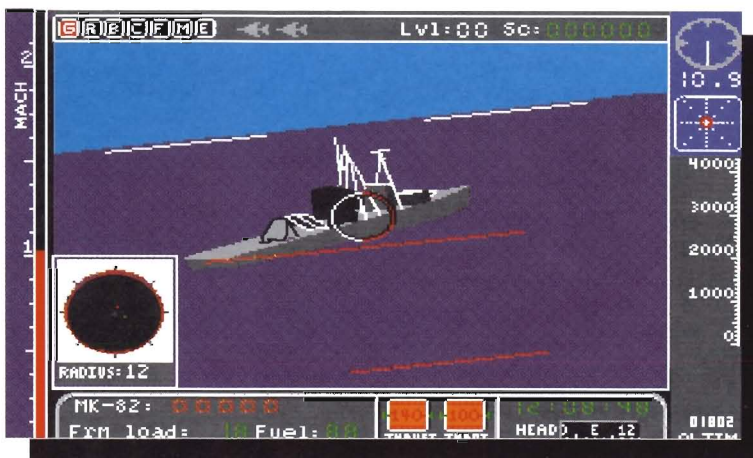
CONCLUSION

The game lacks some features that would qualify it as a true "computer simulator" of the standard Flight Simulator II has set. It comes close, but more work needs to be done, especially with the cockpit display. Despite its weaknesses, Jet is strong when it comes to all-around fun. The action is fast and the graphics are good. □

JET

SubLOGIC Corporation
713 Edgebrook Drive
Champaign, IL 61820
(217) 359-8482
\$49.95

Not copy protected
Requires 512K minimum
Joystick (optional)



(continued from page 57)

The screens you see here were screen grabs captured with another piece of software. The rulers and background grid lines do not appear on paper when a drawing is printed with IntroCAD.

When saving or loading drawings or parts, IntroCAD uses Justin McCormick's fantastic "Pathmaster" file requester. If you've used PIXmate, another product from Progressive Peripherals, or downloaded Justin's "File Injector" program, which he released to the public domain, then you've also seen Pathmaster.

While I may get some arguments on this, in my opinion, Pathmaster is the most powerful, and most full-featured Amiga file requester I've ever seen. Describing it wouldn't do it justice - you simply have to get your hands on it and use it and I think you'll agree.

One of the illustrations depicts the Pathmaster file requester on the IntroCAD work screen.

Pathmaster automatically knows the names of all the devices attached to your Amiga and allows you to navigate around all the directories and subdirectories on any device without ever taking your hand off the mouse. I congratulate Progressive on their decision to implement Pathmaster in more than one product, and I hope it shows up in products from other companies.

PRINTOUT TIME

Ultimately you're going to want to print your drawings, and here too IntroCAD takes a departure from most Amiga graphics programs. IntroCAD doesn't use the PRT: device. It completely ignores all Preferences settings and Workbench printer drivers. Instead of dumping a screen image to the printer, IntroCAD draws a new image with the resolution you select for your particular printer, and then dumps THAT image. A directory full of IntroCAD-specific printer and plotter definitions is included with the program, and only IntroCAD uses these definitions.

While there are hundreds, maybe thousands of different printer and plotter brands and models on the market, most popular printers can "emulate" an-

other brand's model so no matter which printer you own, you should be able to achieve great results with one of the 40 printer definitions included with IntroCAD. There's also a list on the IntroCAD disk which will show you which printers are equivalent to other printers for which a definition file has been supplied.

The printer definition files you choose for your printout will know in which modes of resolution your printer can operate, and present you with a level of choices from "Quick" (fast but not great quality) to "Final" (slow, but highly detailed, dense, and plotter-like) and a couple more between those. My own Epson JX-80 color printer is a fairly standard 9-pin dot matrix job with full FX-80 emulation. IntroCAD's "Final" printout mode gave me very high quality results. If you have an 18 or 24 pin printer, you can expect even better, although I did not have a chance to test IntroCAD with any other printers or plotters.

Here I found one of the few problems with IntroCAD, and that was the Epson JX-80 printer definition supplied on disk does not function correctly. No matter what I tried, I could never get colored areas of a drawing to print in color. Hopefully Progressive Peripherals will check into this and supply a corrected JX-80 printer definition file to IntroCAD owners who require it.

LIMITATIONS

What limitations does IntroCAD have? It has no facility to do "layers" in drawings - that is, all objects you create are on one layer. The objects are all controllable both individually or as groups of objects as explained earlier.

There are no pattern or color-fill tools either. Filling an area with a color or a pattern would require you to draw tightly packed individual lines of that color or a pattern within any given area. But here we're starting to enter a hazy area between what you would use a drawing program like IntroCAD for, or if a paint program would be better equipped to accomplish the goal you have in mind.

As listed on IntroCAD's package, there are many possible uses for a pro-

gram such as this: report illustrations, flow charts (and 18 pre-built flow chart parts are provided on the disk), floorplans, block diagrams, time lines, technical documentation, circuit diagrams, site plans, interior design, and just about anything else you could dream up.

As a "free" bonus, the IntroCAD disk also includes a pair of freely distributable plot generating programs: MultiPlot and ThreeDPlot, which can generate designs based on numeric input which can then be imported into IntroCAD for use in drawings.

SUMMARY

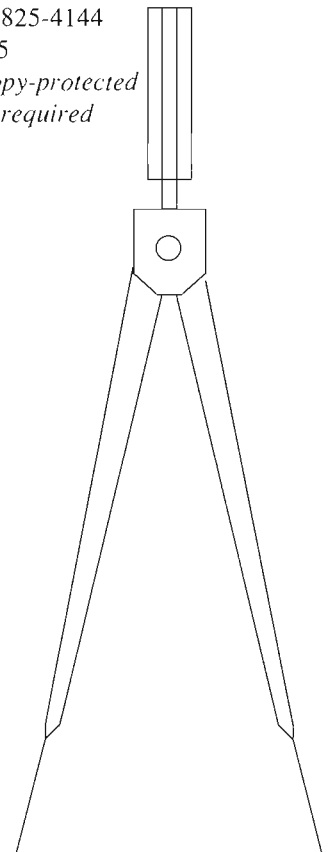
IntroCAD is an economical object-oriented drawing program with an intelligent and "intuitive" user interface, capable of delivering high quality output to any supported printer or plotter. IntroCAD is thoughtfully written and enjoyable to use. □

IntroCAD

Progressive Peripherals and Software

464 Kalamath St.,
Denver, CO 80204
(303) 825-4144
\$79.95

not copy-protected
512K required



The Amiga Monitor

FREE AMIGOTIMES SUPPLEMENT

VOL. 1 No. 1

AMIGA FAMILY TO GROW!

COMMODORE TO EXPAND AMIGA LINE

Plans for the new members of the Amiga family are underway. Commodore is working on three new Amiga computers. The *Amiga 2500AT*, the *Amiga 2500UX*, and last but certainly not least, the *Amiga 3000*. It should be noted that the *Amiga 2500AT/2500UX* are basically bundled A2000's.

The *Amiga 2500AT* will be using the Motorola 68020/68881 processors, an 80286 Bridgeboard, 2 MB of 32 bit RAM, a 5.25" FDD and a 28ms 40 MB autoboot hard drive.

The *Amiga 2500UX* will also be equipped with 4 MB of 32 bit RAM, 68020/68881 and a 28ms 100 MB autoboot hard drive, and will be running version 5.3 of a *UNIX*

compatible operating system. The *UNIX* that will be implemented is a proprietary high performance windowing user shell. A tapestream storage device is optional.

The *Amiga 3000*? Will be based on the Motorola 68030 cpu and use multiple mass storage devices of up to 2.2 gigabytes and have networking capabilities.

At the Hanover CEBIT show, some of the products shown at the Commodore booth included the A2620, a 68020/68881/688851 coprocessor board for the A2000. Also shown were the new high resolution (1008 x 800) Gray Scale Amiga monitor, the A2024, the A2300 Genlock for the A2000, and the A2286 AT bridgeboard, and an Amiga Transputer project.

In closing we have been presented with a bit of late breaking news; by the time this issue hits the

stands, the new Fatter Agnus chip will be close to being available to the public. One major feature of the chip is its ability to address 1MB of CHIP Memory, 2 MB in future machines (CHIP memory is the lower 512K of memory in all current Amiga models). Also based upon the new chip design, there will be the possibility of adding a math coprocessor to the blitter, this can be used to speed up software.

Commodore West Germany is working on a Transputer project which will allow you to run up to 17 transputer chips per Amiga 2000. Bringing the Amigas processing power up to 170 MIPS. A2000's installed with the transputers can be networked together into one single big machine. The transputers run under an operating system called Helios which runs concurrently with AmigaDOS.

GOLDSPELL 2.0 COMING SOON!

By the time you read this, GoldSpell II should be released. It will contain a 90,000 word dictionary which is automatically loaded into ram upon entering the program. You will have the added option of creating or deleting your own dictionary using a wordprocessor, and then loading it into GoldSpell as a file to find those words that do not exist in GoldSpell. If you have several user dictionary files, you can load as many of them as you wish, memory permitting. There is also a function that will give you information on your personal dictionary: like total number of words, total space they occupy in memory and the number of new words accepted since the program started.

GoldSpell allows you to check an entire text file imported from a range of different wordprocessors (from ASCII to Wordperfect) and you have the option to quit the spell checking at any time. Another neat feature is batch spell checking, which goes through an entire document without interruptions while composing a list of errors (addressed at your convenience). There is also a feature that allows the spell checker to ignore certain words from its dictionary. Once all the spelling is corrected you can test the readability grade level of the program using the gunning Fog Index or analyze the document for word frequency usage. Updates from version 1.0 will cost \$20.00 U.S. (\$25.00 CDN).

GoldSpell II
Gold Disk Inc.
P.O. Box 789, Streetsville,
Mississauga, Ont. L5M 2C2
CANADA
(416)828-0913
Not copy protected
\$44.95 US \$59.95 CDN
512K Required

C LTD. FORMS CONSORTIUM

C Ltd has initiated the most ambitious consortium project for the Amiga to date, all of which is slated for introduction at the spring Comdex show in May, with the products commercially available by the 1st of June. C Ltd is trying to put the Amiga into the "small business market" by providing networking capabilities (multiple-user access) for specific hardware and software. They have developed a software package capable of allowing networking for their SCSI Host/Controller to all other SCSI device hardware. Specifically, this software will be used to link up three new hardware products that C Ltd is rushing to complete.

C Ltd will also be releasing a SCSI-based 300 DPI laser printer, for under \$2500. The wait time for output is ten seconds times the number of users currently trying to print. This fast printing is possible due to the Amiga pre-processing the data into a bit-map and then transferring this data to the printer via a "1 byte wide, high speed SCSI bus" in one

continuous eight second burst. The second piece of hardware is a SCSI 300 DPI page scanner, for \$1500. The scanner scans a page of data and directly prints the initial copy in less than forty seconds, with each additional copy printed at a rate of six pages per minute. Lastly, a 9600 baud modem card based on a Zorro bus, with FAX transmission capabilities, priced for under \$600.

C Ltd will also be bundling third party software packages with various hardware items. So far the other companies involved are: General Computer Corp., supplying a version of their "Personal Laser Printer" which contains the latest generation Ricoh laser engine.

SoftLogik Corp. will be supplying a version of Publishing Partner Professional (desktop publishing software). It will be able to import scanned images, overlay existing pages of scanned information with formatted text and graphics, create formatted pages which can be directly printed, overlaid or edited.

Professional Automated Re-

sources Software Inc. will create a version of Express Paint that will allow bit level manipulation of pages before or after printing.

Avant-Garde Software will be supplying the custom amiga preferences printer driver that will allow most programs to make use of the laser printer.

Associated Computer Services Software Division Inc. will be supplying the scanner driver software, which will allow the creation of either IFF data files in various resolutions or full page 300 DPI bit mapped pictures.

Soft Circuits will be supplying a version of PCLO (Printed Circuit Layout), and QCAD (CAD software).

C Ltd.
723 East Skinner
Wichita, KA 67211
USA

TURBO SILVER MODULE FOR INTERCHANGE

There are a number of great animation programs presently available; Sculpt 3-D, Videoscape 3-D, and Forms In Flight, to name but a few. But how can you use the best features that each animation program has to offer? You might try using Interchange. This program provides a full intuition interface for all its functions, but it is not a stand-alone animation program. A new module has been released for Interchange. This new conversion module allows you to use Turbo Silver to ray trace Videoscape 3-D objects.

The master program plus Sculpt 3-D and Videoscape 3-D conversion modules is \$49.95. Forms In Flight add-on conversion module is \$19.95 and the Turbo Silver add-on conversion module is \$19.95. Object disk #1 includes Sculpt 3-D and Videoscape 3-D objects and is also priced at \$19.95.

Interchange Syndesis

20 West street
Wilmington, MA 01887
USA
(617) 657-5585
\$49.95
\$19.95 (each add on module)
512K Required

IT'S A WORD, IT'S A PLANE, IT'S DESIGNTXT!

NEW WORDPROCESSOR

At press time, we had not yet received the final version of *DesignText*, however, we were very impressed by the demo version we had. It's something amazing, something new and it's not just another word-processor. Simply put, it has everything and - unlike other word-processors - it's scrolling and screen refresh rate are very fast, even in hi-res mode. The screen can open up into overscan mode and support horizontal scrolling of up to 255 characters. Don't be fooled by the row of icons at the top of its screen, this is not a "kiddy" word-processor, it's just very user-friendly.

All standard text editing commands are available, including a full-featured search function (forward, reverse, local, global, case sensitive and case insensitive). You can insert headers, endnotes, create a table of contents and an index. The "Page Setup" option actually shows where each pages text begins and ends. Multiple fonts are available, including special fonts like Greek, Math, Cyrillic and other special symbol fonts.

Loading documents from the disk is a fairly quick and easy pro-

cess. Several windows can be opened simultaneously, each one loaded with either an existing *DesignText* format or text written previously on another word-processor. *DesignText* also allows the importation of IFF graphics onto text documents. How about inter-window text transfers? No problem with *DesignText*.

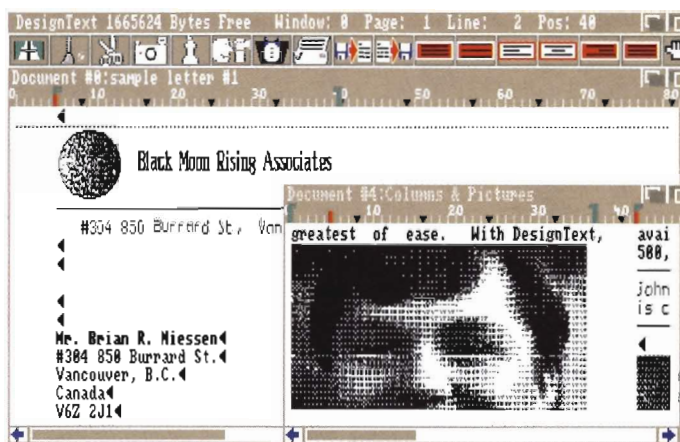
From here on in *DesignText* goes beyond the capabilities of the

100,000 word dictionary and an interactive database (called *PeopleBase*) are also included. *PeopleBase* allows you to retrieve needed records and/or data from your database and insert it directly into your current document. *PeopleBase* can also act as a mail list manager and be used to print mailing list labels. Did we hear somebody say, what about keyboard macros? Its got that too; you can set up macros

for twenty-six separate phrases and ten separate printer code settings. Then when all this is done you can delete files or format disks from within the program. This is not all, there are simply too many features to describe.

This may just be the word-processor that will set new standards. We

are eagerly awaiting the commercially released version in order to provide you with an in-depth review, so stay tuned to AmigoTimes.



An example of *DesignText*'s multiwindowing capabilities.

common word-processor and tackles the exceptional processors category rather impressively.

First of all it will be available in English, French and German. It will allow the creation of tables using multiple columns; which can be sorted, averaged, and then calculated for percentages, all within the processor. There is also an on screen calculator and an auto-backup function that saves your text at a set time interval. A built-in outliner (a.k.a. an idea processor), a

DesignText
Designtech Business Systems, Inc.
850 Burrard St., #304
Vancouver, B.C. V6Z 2J1
CANADA
\$129.00

"HOLY SMOKES BATMAN IT'S COMICSETTER"

Gold Disk's long awaited *ComicSetter* will be available at the end of June. If you're an aspiring comic artist trying to fulfill your dream of one day publishing your own comic, this may very well be the program for you.

Comic Setter is designed to run on a 512K Amiga, but 1MB is recommended. Each page is composed of individual boxes (better known as "panels"), which may be reshaped to any size. Panels may contain multiple layers of images which are combined with intelligent masking (this renders individual shapes with their own identities), and automatically clipped to the panel's border.

The user interface is set up in a Desktop Publisher manner allowing use of grids to aid in comic layouts, and several levels of magnification for doing detailed work. Grouping operations allow manipulation of multiple graphic elements, as single units. You can choose from four page sizes or you can design your own up to 8.5" x 14"; a variety of line weights and colors are provided, as well as standard paint program functions such as, area fills using both solid and multicolor dither patterns. A variety of brush shape and sizes are available, along with a powerful airbrush with adjustable "spray distance" and "ink flow".

Bezier curves, and polygons are supported in both bitmap and object graphics.

Display resolutions are hi-res interlaced, or square-pixel medium-res mode with the ability to import bitmapped IFF ILBM format graphics. In the area of text, *ComicSetter* will have special comic book fonts, providing left, right, or center alignment, and a variety of predefined balloon styles that may be fully edited. All Amiga fonts are supported along with the various text styles, bold, italics, outline, and underlined. To output your masterpieces, all printers in Preferences are supported with user-specified DPI

(Dots Per Inch), for the best output quality.

The main disk will retail for \$99.95 US (\$134.95 CDN). Three clipart disks (Funny figures, Super Heros, and Sci Fi) are each sold separately for \$34.95 US (\$45.95 CDN).

Comicsetter
Gold Disk Inc.
P.O. Box 789
Streetville, Ont. L5M 2C2
CANADA
(416) 828-0913

SPELLCHECKING WITH ZINGSPELL!

Zing!Spell is a spell checker that can function while you type (or check an already existing article) and will beep or blink every time a typo or spelling error is made, this feature can be turned off if it annoys you. Its 95,000 word dictionary is memory residing, but if your memory situation is tight the spell checker can still be used from the disk. There is also the capability of adding a custom dictionary. The interactive batch spell checker works with most of the major word processors.

Zing!Spell's main screen opens up displaying the last misspelled word and total number of words that were misspelled. Fully opening this window allows viewing of all misspelled words from which a custom dictionary can be created. The rest of the features of the program are accessible at the bottom of the main screen in the form of three gadgets.

HW or the spelling help window gadget displays a misspelled word, along with suggested correct spellings, from which you can choose the right one.

EW or exchange window, allows repetition of lengthy words or sentences (up to 120 char long) by typing in a designated string. This window makes use of two string

gadgets; "Change From" specifies string to search for and "Change To" specifies the replacement string for the search string. The amount of search strings is limited to the size of your memory.

OW (options window) offers several spell checking modes: check spelling after each word, after each line, anytime while writing or check last word entered by pressing a special hot key. Your custom dictionary is also manipulated from here and is limited to the size of your memory.

Also from within the OW or EW windows a Directory Window is accessible, which simply allows you to save or retrieve words from the disk.

The price of *Zing!Spell* is \$79.95 or send in your original spellchecker with its manual and pay only \$39.95. Future *Zing!Spell* update will contain a thesaurus and a CLI syntax checker, but best of all it will be free!

Zing!Spell
Meridian Software Inc.
9361 West Brittany Avenue
Littleton, CO 80123
USA
\$79.95

PROWRITE 2.0!

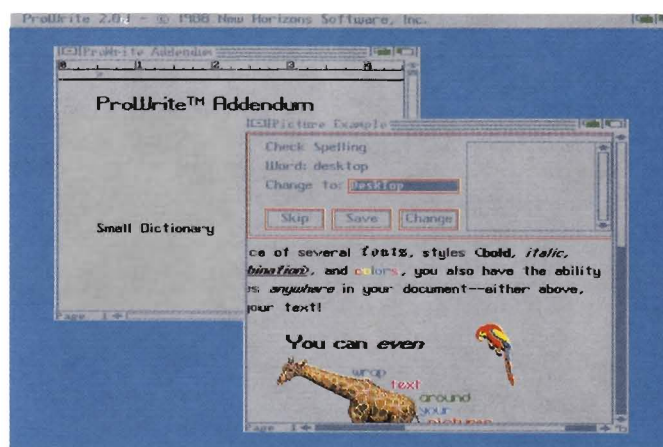
ProWrite 2.0 is an improvement over 1.0. It's still a WYSIWYG (What You See Is What You Get), multi-font, color graphics, word processor. Now it sports a 95,000 word spelling checker, allows you to re-size IFF pictures, provides HAM support and works in over-scan. Also, paragraphs in your text can now be sorted and this word-processor's defaults can be user defined. This new version contains the new Workbench 1.3 printer drivers.

ProWrite 2.0 is still a memory hog in hi-res, so if you only have 512K use med-res. *ProWrite 2.0* is a

good word-processor to use, especially if you want to incorporate graphics, and if your machine has more than 512K of memory.

Trade up to ProWrite offer!
Mail in your present wordprocessors master disk and get *ProWrite V2.0* for only \$75!

ProWrite 2.0
New Horizons Software
P.O.Box 43167
Austin, TX 78745
USA
\$124.95
512K Required (1Mb recommended)



ProWrite 2.0 displaying WYSIWYG environment.

ZOOM! A STORY OF ADDICTION

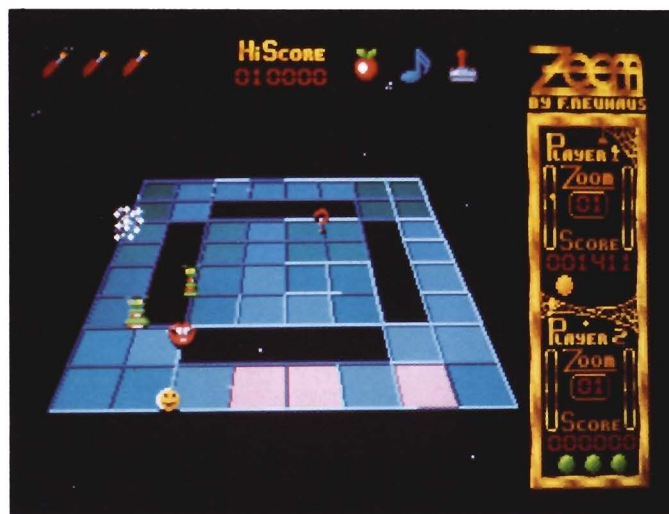
ZOOM! is definitely one of the best arcade-style games available for the Amiga. Right from the beginning of the game, *ZOOM!* dazzles you with its incredible graphics, sound effects, and music. The opening animation sequence, a stage featuring a campy vaudeville act, is hilarious. The game takes place on a three-dimensional grid in space and you move a pacman-like character called "Zoomer".

The object of the game is to race around the grid encircling squares and filling them with colors. Once all of the squares have been filled, you advance to the next level (there are fifty levels and none of us have been able to get past the first few). Helpful objects such as magic potions and time freezers appear throughout the grid and help you avoid odd enemies such as aliens and, get this, a pair of lips. You can play alone or with a second player.

ZOOM! requires at least one joystick and can be played on any Amiga.

At this point it is only fair to warn you that this game is incredibly addictive and therefore a perfect candidate for installation onto your hard drive at the office. Unfortunately *ZOOM!* runs under its own DOS this means you can't run it from Workbench or copy it to your hard drive, this we find is its only drawback.

ZOOM!
Discovery Software International
163 Conduit Street
Annapolis, MD 21401
USA
(301) 268-9877
Copy protected
No tentative price
512K Required



ZOOM! from Discovery Software International.

RELEASE OF NEW AND EXCITING VERSION OF ARKANOID!

Discovery Software International has just announced the release of a new version of their highly praised arcade game Arkanoid. Arkanoid was converted from Taito Corporation's 1986 best selling video arcade game. This 66 level (33 new levels have been added to the original 33) game has a special feature for the disabled and a 30-day unconditional money-back guarantee.

The programmers at Discovery Software are particularly excited about a new feature in this version that can slow the action so players can enjoy the game at a pace they can handle. This change is expected

to make the program enjoyable to a large group of people to whom video games were previously inaccessible; Discovery Software is hoping that other game manufacturers will follow suit with this feature.

Discovery Software has also announced a price reduction for Arkanoid in the U.S. and Canadian markets, effective May 15.

Arkanoid

Discovery Software International
163 Conduit Street
Annapolis, MD 21401
USA
(301) 268-9877
\$29.95 U.S. (Previously \$49.95)
512K Required

PROFESSIONAL TERMINAL EMULATION

Progressive Peripherals & Software Inc. is now shipping their professional terminal program, *Dr. TERM Professional*. This is a package for the Amiga 500, 1000, and 2000

computers. Some of the features of this package are Amiga to mainframe communication, a 300 entry phone directory wherein each phone book entry may have custom communication parameters.

Dr. Term Professional features an extensive command language for programming up to 40 different macros using the function keys;

macro files can be called from configuration files. Users can define and execute a series of commands using the Script Language that is included with the package. Another feature of this package is a "Remote" mode with a multiple level password which acts like a mini-BBS. In the area of terminal emulation, Dr. TERM Professional's emulations include: VT100, VT52, and TTY. Commands may be made with the mouse or keyboard equivalents. The manual is extensively documented, and designed to cover any on-line eventuality. Communication protocols include XMODEM, YMODEM, and WXMDEM transfers.

Dr. TERM Professional Progressive Peripherals & Software, Inc.

464 Kalamath Street
Denver, CO 80204
USA
80204
(303)825-4144
\$99.95 US
512K Required

SHAREWARE

This Column will be devoted to the announcement and review of shareware and PD. AmigaTimes invites developers of shareware to submit their programs for review in this column.

Thuman Programming disks claim that they allow the user to program their unconscious mind, change their unconscious self image thereby becoming what they want to be. Thuman Programming disks use "Hemi-sync" (a patented auditory guidance system), and "Affirmations" (via words going across the mouth of a "happy face") to help realign attitudes about specific behaviors.

To achieve the full effect the programs require that stereo headphones must be connected to the Amiga while using the program. It should be noted that this program is recommended for adult recreational use only. The programs are written in Basic, and a range of different self image disks are available. The disk that we had the opportunity to try was titled *Slender & Sexy*. All of the disks in the series are shareware. For further information contact:

Thuman Programming
Route 1 Box 175-G
Faber, VA, USA
22938

INTERCHANGE™

Share objects between Sculpt3D, VideoScape 3D & Forms in Flight

NOW YOU CAN...

- Use Sculpt 3D to ray-trace VideoScape 3D objects
- Do Forms in Flight animations on Sculpt 3D objects
- Create VideoScape 3D objects using the Sculpt 3D interface

Full Intuition interface for all InterChange functions

Object Disk #1 now available. Includes a Sculpt 3D font, plus lots more Sculpt 3D and VideoScape 3D objects!

InterChange master program plus Sculpt 3D and VideoScape 3D Conversion Modules, \$49.95. Forms in Flight add-on Conversion Module, \$19.95. InterChange Object Disk # 1, \$19.95.

This product requires objects from Sculpt 3D and/or VideoScape 3D and/or Forms in Flight. It is not a standalone animation program.

To order, send check or money order. Please include \$3.00 postage & handling. MA and WI residents add 5% sales tax. InterChange is a trademark of Synthesis. Sculpt 3D, VideoScape 3D and Forms in Flight are trademarks of Byte by Byte Corporation, Aegis Development and Micro Magic respectively.

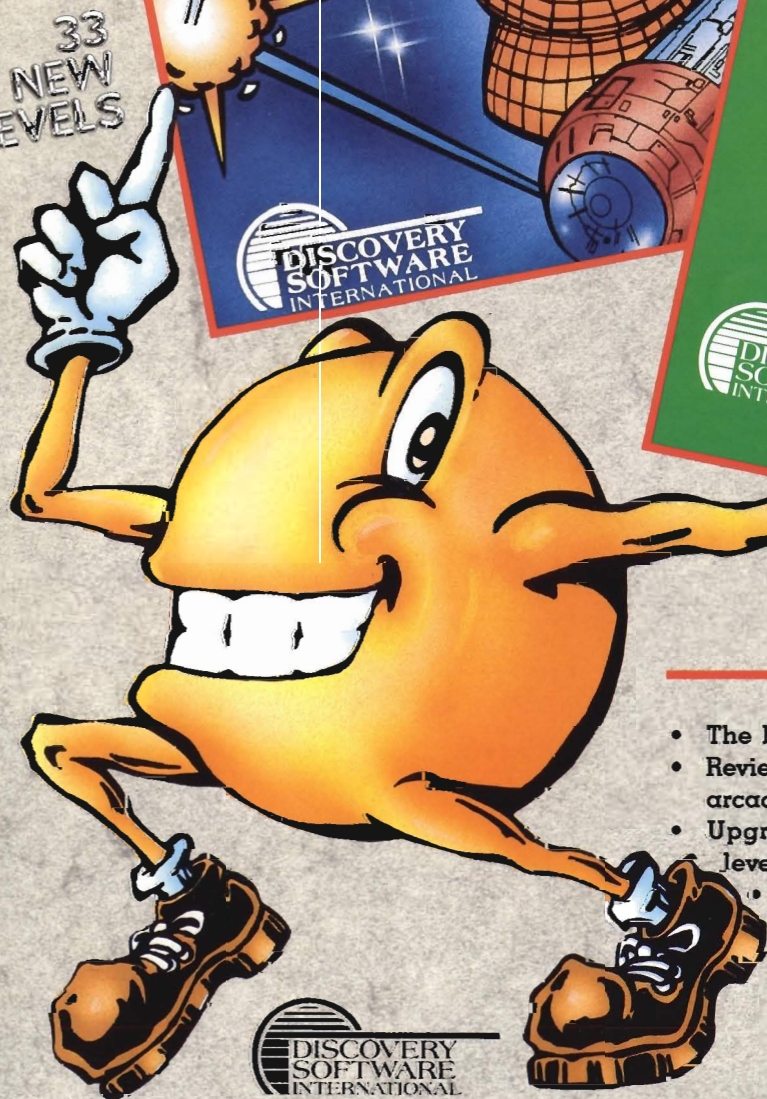
SYNTHESIS
20 WEST STREET
WILMINGTON, MASSACHUSETTS 01887
617-657-5585

Ad Index

Advertiser	Page No.
CSA	Inside front cover
DesignTech Business Systems	27
Discovery Software International	Inside back cover
Electronic Arts	22 - 23
Gold Disk	Back cover
Jupiter Technologies Inc.	50
MicroIllusions	41
Microway	5
Mindware International	49
NewTek	6
Synthesis	64

ONE GREAT GAME

DESERVES
ANOTHER



ARKANOID

- The best just got better
- Reviewers agree: unparalleled arcade quality
- Upgraded to 66 knockout levels!
- Twice chosen as arcade "game of the year"
- 30-day unconditional money-back guarantee

ZOOM!

- At last, a dynamic new game concept
- A mesmerizing combination of action and strategy
- 50 levels. 1-2 players or 2 players at once
- She'll love it too, so will the kids!
- 30-day unconditional money-back guarantee



Discovery Software International products are available at your favorite retailer.
Call 1-800-34-AMIGA for details. Arkanoid, \$29.95. ZOOM! \$29.95. Prices may vary.
Discovery Software International, Inc., 163 Conduit Street, Annapolis, MD 21401. 301-268-9877.
Arkanoid and ZOOM! are compatible with any Amiga™ model. Amiga™ is the registered trademark of Commodore-Amiga, Inc.
Taito™ and Arkanoid™ are registered trademarks of Taito America Corporation. ©1987 Taito America Corporation.
All rights reserved. ©1987 Discovery Software International, Inc.



GOLD DISK

COMIC SETTER

LUSCIOUS
COLOR!

KILLER
COMIC
FONTS!

WOW!
WORD
BALLOONS!

CARLOADS
OF CARTOON
CLIP ART!

BOFFO
BIRTHDAY
CARDS!

IT PUTS
THE POWERS
OF THE PROS
AT YOUR
COMMAND!

KNOCKOUT
NEWSLETTERS!



YOU CAN CREATE YOUR OWN MULTI-PAGE COMIC BOOKS WITH ComicSetter - THE COMPLETE COLOR COMIC DESIGN STUDIO FOR THE 512K AMIGA. WITH STRUCTURED AND BIT-MAPPED GRAPHICS. ONLY \$99.95 FROM GOLD DISK SOFTWARE. SEE YOUR LOCAL AMIGA DEALER, OR CALL 1-800-387-8192 TO ORDER.