Monkey Island II: LeChuck's Revenge

by Paul Miller

Lucasfilm Games has done it again. I'll just skip any corny introduction I might have spent more effort cooking up -- This game is simply amazing!

Guybrush Threepwood is back, a little older, a little less naive, and with an entirely new adventure ahead. Monkey Island II has four islands, an evil pirate's fortress, some new friends, very tricky puzzles, really cool music, and that great Lucasfilm humor. All on a whopping ELEVEN disks! You don't need a harddrive, but if you don't have one here's your excuse.

You don't need to have played the first Monkey Island, but it helps, since you'll meet up with people from the first game. Ever wonder what happened to 'ol Stan, or how Guybrush fared with Elaine?

Now Guybrush is after the legendary treasure of Big Whoop, which could also be his ticket to finally destroying his arch-nemesis and really creepy dead dude, the Ghost-Pirate LeChuck! That's right, LeChuck is back too and revenge is on his decomposing mind.

Better brush up on your voodoo and roulette, because this one's a toughy. Fortunately, there are two versions of the game: the normal version and the "whimpy" one, which doesn't have all of the puzzles. Some of these brain-teasers aren't very obvious at first (or even last), but once you figure them out (or get some nice person on the Internet to give you some hints) you'll likely laugh your head off! The frustration and anxiety are well worth the effort!

If you're not familiar with this type of game, it's what's commonly termed an "interactive graphics adventure." Most of the screen is the view of what's going on in the game and the lower portion of the screen has a bunch of words describing what you can do, along with little icons representing your inventory. To look at an object, you would click on "look" and then on the object (either on the viewscreen or in your inventory). There are keyboard shortcuts for most of the commands, so you can press 'L' and then point the mouse at the object. A two-handed operation like this works a bit faster than just using the mouse. When you need to talk to another person, the control area turns into a list of phrases you can say. Click on a phrase and it appears above Guybrush's head while his mouth moves (as do all characters when they speak). He even gestures with his hands convincingly where appropriate! At any rate, the interface is very intuitive and easy to learn.

The graphics were converted very well from 256-color VGA. The animation is fluid, and most of the sequences are quite humorous. Sound effects are limited but the music is great, changing styles depending on the location, and setting a distinct mood for each. It's especially nice running through a sound system with a bit of hall reverb if you have a digital signal processor.

The game will run in 1 megabyte of RAM, and the copy protection is a cute code-wheel affair with some very interesting cures for some common pirate diseases. There's only one problem: it won't run under AmigaDOS 2.0. You have to switch to 1.3 to get it to work correctly. This may be a problem for A3000 owners with 2.0 in ROM. Perhaps some tweaking with cache and RAM options will allow it to work.

Retail: $59.00

I can't rave enough about Monkey Island II. What an experience! I spent a whole week on it and hated when it was over -- and what a surprise! I can only hope there's more to come from LucasFilm Games.

On a scale of one to ten,
this is a definite TEN!

BUILDING A WORLD

SimEarth vs. Global Effect

by Mike Neylon

Remember the good old days of Populous and SimCity? Two relatively simple games to learn, but required strategy and planning to conquer? Well, with the release of SimEarth and Global Effect (as well as Civilization), you can now prepare yourself for hours of strategy and planning as you construct your own world.

SimEarth, by Maxis, uses the same premise as SimCity and SimAnt: the development and manipulation of 'sims', computer-controlled creatures that follow basic instinct. Your goal in SimEarth is to create a suitable environment for your sims to evolve from singled-cell life-forms to space-age civilizations. You are allocated a 'energy budget' which grows based on the number and type of life-forms present. Your energy is reduced as you perform actions on your planet: adjusting the water cycle, increasing seismic activity, decreasing the reproduction rate, and several other features. You can also add life-forms and terrains to the planet. And for fun, you can send that random meteor or tidal wave and see how the sims react.

The full game is played in 4 stages. Each stage represents an evolutionary cycle, and requires a fulfillment of a goal to proceed to the next stage. For example, in the first stage, you must create a planet suitable for life, cause life to begin, and to get land-roming creatures. In the final stage, you must take the sentient race from the start of the Industrial Age to the beginning of space travel. The time scale in each stage varies, from billions of years in the beginning, to hundreds of year at the end.

The sims, as well as the planet, are controlled by a theory known as the Gaia effect. The theory states that the planet and life are connected in complex ways, and the actions of one will affect the other. This is very well implemented in SimEarth. Once cities start appearing on your planet, you will notice an increase in temperature, carbon dioxide levels, and the planet will react, with severe weather and changes in the environment. A special interface can be used to 'talk' to your planet, but this merely consists of a few sentences of advice.

Global Effect, from Millennium, also uses the concept of creating your own world, but from a different starting point. In Global Effect, you must colonize a world, using its resources wisely and carefully, but without upsetting the environment of the world. Again, you have an 'energy budget' which grows with the production of fuel from the planet. Building production sites, cities, forests and park lands, and farms all cost money. However, you are less directly in control of the planet. If the planet starts to freeze, you just can't increase the greenhouse effect - you must build services to fix that, or destruct the existing services that create the cold temperatures.

Power can come from several resources including oil, coal, nuclear, solar, and wind. If necessary, the resources must be processed before use, and storage devices must be created. Power must be supplied to the city, and distribution is very important. You also must provide fresh water, sewage treatment, recycling programs, and hospitals for the people. However, an eye must be kept on resources and the planet. Once mines are mined out, they are shut down, and are unusable.

Several one player situations are available, including creating a colony on several different types of planets and repairing damage done to a planet. However, there are also features to play with a second player over a modem, or to play against the computer, either to create a world, or to win control over a world. Several services are added for that goal, including missile silos, airfields, and harbors.

Global Effect is filled with information, as is SimEarth. Ozone layer levels, pollution, and contamination are just some of the data one can inquire about. Both games are HD-installable, and work under both 1.3 and 2.0. Neither require a lot of memory (can be run on a 1 meg machine), and both are multi-tasking.

I must admit, Global Effect has one of the best-looking interfaces I've seen for a game. The 32-color presentation is magnificent, and can be played in NTSC or PAL. SimEarth is only 16 colors, but some of the creature icons are done quite well. The worlds in Global Effect are much larger than those in SimEarth, and seem to have a much better resolution in the end. The SimEarth interface also seems to have a few bugs (a button doesn't activate once in a while). However, control in SimEarth is much easier. You can select an object using a menu-like technique, while in Global Effect, you must scan through a list of icons to choose an object (you can define the function keys to select an object). Moving in SimEarth involves moving the mouse to the edges of the screen causing it to scroll, while in

AMIGAphile survey continued...

scrap metal. I know that piracy exists but all the problems can't be blamed on that alone. Among pirates there is a saying, "If it's good enough to take up disk space, it's good enough to buy". I do wish companies would do away with their stupid protection schemes. Pirates will always get around them and they annoy the helpless user who owns a legitimate copy. That's enough preaching. Think about supporting the software developers. They will be the ones to say when the Amiga is through in the market. Even if Commodore stops making them, there will still be those who have one.

HARDWARE HACKERS

The Amiga was originally 'the' hacking machine. Many users do their own hardware upgrades. A1000 owners are particularly prone do solder things inside their system. 22% of those surveyed do, or have done, their own hardware hack. The number of successful hacks was not tabulated by this survey. Many of them are simple and most are the only way to upgrade an old system. I'd be more willing to take a soldering iron to my machine if I knew I could upgrade (read A1000 owner).

AVERAGE USE

How often do you use your Amiga? Now that I do this newsletter, my Amiga only gets turned off long enough for me to sleep. I do everything on my Amiga (except fry eggs - even though the SCSI drive does run hot enough).

24 hours a day	19	*******************
More like 12 hours	23	***********************
Once a day	31	*******************************
Five time a week	9	*********
Three time a week	9	*********
Once a month	2	**
I know it's here...	1	* [I think he said it was broken]

WHAT I DO WITH MY AMIGA

I tried to include a selection of the most common things done with a home computer (especially the Amiga). Two people threw me a curve and said they use it to teach. Now that's refreshing news. Since most of the surveys came from people on InterNet, the large number of people doing telecommunications is expected. This gives AMIGAphile the large reader base. Everybody is a game player. The Amiga has the talents and is often necessary to blow of some steam. The rest of the figures, I leave for you the decipher. I wonder what the software market is geared towards...

Telecomm	74	**************************************
Game playing	66	**********************************
Word processing	64	*********************************
Programming	64	*********************************
Graphics	49	*************************
Audio	35	******************
Animation	25	*************
Desktop Publishing	19	*********
Video work	15	*******
Spreadsheet	9	****

POPULAR PROGRAMMING LANGUAGES

It seems that C is envogue on every platform but Amiga users are entrenched in that camp as well. The Amiga always seemed very suitable for programming in C and Commodore seems to encourage this buy looking at their Programmers Reference manuals. The ever classic BASIC holds second place but I think this is because of the efforts to make it more structured. When I was growing up, everyone knew BASIC. I guess, learning new languages is never fun and with the improved environments so BASIC packages offer, there is little incentive to change. I expected assembly to rank high on the list since many things are done better using assembly language; even if it is in-line code in a C program. The rest of the contenders are languages which cater to a specific programming task. Being at a university gives one the opportunity to be exposed to many different languages but I don't imagine too many companies have accepted some of the newer languages. Maybe, it's because many of these languages aren't support by the big names in language packages.

C	64	*********************************
BASIC	22	***********
Assembly	18	*********
ARexx	12	******
Modula-2	9	****
LISP	8	****
Fortran	7	***

WHAT OTHER MACHINES DO AMIGA USERS OWN?

Sixty one percent of Amiga owners have another system (two timers, hmpfh). Forty three percent own an IBM or a compatible. This doesn't surprise me for some reason (I own one too). Windows is a face-lift for a poor excuse of an operating system. Excuse me, Bill Gates, you need how much RAM to multi-task? 4 meg minimum? I think I can name that tune in 512K. Most IBM users are zombies who do what they're told without know why. They say you need a 386 or better, then that's what you buy. I must admit that the prices are low and the graphics are superior (for now). Macintosh owners make up 14% of those who own another system. They are easy for novices to use. They don't have any of that Guru nonsense the early Amigas seemed to proliferate. The other 40% own something different. These systems range from Commodore 64's to Apple]['s and Atari's. Most of these systems just collect dust. I know, I have one of each. I just like to have the latest and greatest. I guess that's why I didn't buy a Nintendo (needed more computing power).

HOW'S OUR DRIVING - COMMODORE

Oh boy did I receive a ton of comments on this question. Everybody has an opinion about Commodore. At least we are entitled to this one. The categories to rate were marketing, sales, support, reputation, and overall opinion. These are all opinions and reflect the feelings of Amiga owners everywhere. Commodore (tap tap tap) are you listening? This could save you company... (trailing off into the hot air). Commodore has exhibited very little marketing. Take some initiative. I just got a packet in the mail about the Amiga as a multimedia machine. It's great literature. Now, put it in some magazines (I got this by filling out a reader service card in Byte). The figures indicate that people feel sales have been going well but not great. Mediocre... the computer isn't and the users aren't ... who does that leave? Support was rated at Good. This is encouraging. I find support to be very important and would like to take the time to thank everyone from Commodore who participates in discussions on the InterNet. Commodore's reputation ranks at average. Is this how the owners see Commodore or how other see Commodore? I don't know (best Pee Wee Herman impression). Overall, they rate better than hangnails but worse than pepperoni pizza. Now if we could just get some really good press...

	Great	Good	Average	Yuck	Seriously?
Marketing	1	1	9	29	48
Sales	3	16	56	10	1
Support	1	37	28	3	5
Reputation	1	12	33	31	11
Overall	0	25	31	19	12

RIDING INTO THE SUNSET

Most of all, this survey was fun. I got to introduce myself to many different people and find out if AMIGAphile would be worth the effort. The numbers never lie and I think the questions where pertinent. I hope you learned something. Y'all come back now, ya hear!

The next section is the actual code, as you can see from the line that starts 'section mycode,code'. I include the section directive because (1) I come from the MS-DOS world where it's MANDATORY to have sections for code, data and stack (old habits die hard) and (2) because I tend to write large programs that span several files, and this way, the linker (Blink) will string together all the code sections that have the same name, and all the data sections with the same name. It keeps things more manageable with small files than one large monolithic piece of code that's about 70K in size!

Sections are optional under A68k, and you can forego them, but I feel that the benefits of using this method outweigh any other reason you may have for not doing it. Anyway ... on to the actual code.

The first thing you'll notice is that I open up dos.library. Since this is a complete stand alone program (not linked with anything else), I have to do this step. The next thing I have to do is get the file handle for output to the CLI window (usually, unless you re-direct output to a file, in which case, the file handle I get references that file). Only then can I actually output any message.

After the message is written, I then can close the dos.library and then simply return to the system.

Wow! Quite a bit of code just to print out a 13 byte message. Well, welcome to the wonderful world of Assembly. One thing you'll notice is that the source code for programs written in Assembly tend to be a bit larger than their counterparts written in higher level languages (HLLs) like C or Pascal, but the resulting executable is usually smaller (unless you link in a large static library like c.lib). Because with Assembly, you're more or less telling the CPU exactly what you want done, as opposed to, say C, which has to take a more general route to get something done.

So, track down an Assembler and linker (maybe even an editor), read your 68k reference manual, peruse the RKMs (if you have them) and play around with the code some, and next month, we'll see about getting rid of those nasty gurus that seem to keep popping up on the system, if you're like me and still running under 1.3. Even if you have 2.0 (which, if I understand, has done away with gurus), you will still find next month's topic interesting. Also, some code to output more than just static text.

See you later ...

*   *   *   *   *

About the author: Sean Conner is a 23 year old student at Florida Atlantic University well on his way to becoming a tenured undergraduate. When not pretending to work, consulting on programing under MS-DOS, reading news, hanging out with friends, sleeping or eating, he programs the Amiga. He may be reached via Internet at 'spc@pineal.sci.fau.edu'.

AMIGAphile Survey - part III: The Final Chapter

Overall, the AMIGAphile survey was fun and enlightening. I hope to do this again next year but until then, here are a few final figures to think about.

SHAREWARE

Eighty three percent of those surveyed (chose Trident for their patients who chew gum) use shareware programs. The Amiga has the largest number of shareware programs available. This makes 83% no surprising. Most users would never consider using software without paying for it. This would ensure that developers keep developing and a good software market would exist. The problem is that only 56% register the shareware they use. At least 20% register some of the shareware programs they use. They are honest about this fact. Some authors don't support their products and I think users should demand good support before registering. The bad news is, this leaves 24% who don't register the shareware they use. Bad user! Bad user! The above figures are alarming. Everyone complains about the software quality, the product support, the lack of upgrades, blame it piracy. Software companies are dropping the porting of their software to the Amiga. This is good news for those who have ugly, slow, IBM ports but this is bad news for the software market. The IBM market is more lucrative. If you use the software, register it. If there is a reason you won't register it, let the author know. Without software all our fabulous hardware is

continued on page 7

```
;-----------------------------------------------------------------------------------------

        include "exec/types.i"
        include "exec/funcdef.i"
        include "exec/exec_lib.i"

        include "libraries/dos_lib.i"
        include "libraries/dos.i"

;-----------------------------------------------------------------------------------------
;*****************************************************************************
        section mycode,code

start       move.l  #doslibrary,a1      ;open DOS library
            moveq   #0,d0               ;any version is fine
            move.l  4,a6
            jsr     _LVOOpenLibrary(a6)
            move.l  d0,dosbase          ;save DOSBase
            beq.w   clean_exit          ;if NULL, exit program

            move.l  dosbase,a6          ;Now get handle to stdout
            jsr     _LVOOutput(a6)
            move.l  d0,stdout

            move.l  stdout,d1           ;output 'Hello world!'
            move.l  #hello_text,d2
            move.l  #_s_hello_text,d3
            move.l  dosbase,a6
            jsr     _LVOWrite(a6)

clean_exit  move.l  dosbase,d0
            beq.s   clean_exit99
            move.l  d0,a1
            move.l  4,a6
            jsr     _LVOCloseLibrary(a6)
clean_exit99        rts

;-----------------------------------------------------------------------------------------
;*****************************************************************************
        section mydata,data

dosbase         dc.l    0
stdout          dc.l    0

doslibrary      dc.b    'dos.library',0
hello_text      dc.b    'Hello World!',10
_s_hello_text   equ     *-hello_text

;*****************************************************************************

        end     start
```

Hello World sample code outputs 'Hello World' to CLI window

AMIGA ASSEMBLY

An Introduction - 'Hello World'

by Sean Conner

Welcome to the first of a (hopefully) ongoing series of articles on Assembly Language for the Amiga.

Anyway, after some thought, I decided to stay away from the traditional course of things, where I spend an article describing binary, octal, decimal and hexadecimal notation, then spend several articles describing the instruction set of the MC68000 (shorthand version: 68k), and THEN spend an article on how to use an Assembler and linker and then...

I've had enough of that type of teaching method in class. Besides, there are many, many books out there that can teach you that stuff. Like I said, I'm not going the tradition route here.

"Well," you say, "if I can just buy a book and learn Assembly Language that way, why waste my time reading this?" Glad you asked (pretend you asked). Most books on the subject (about 99%) on Assembly go into excruciating detail about the instruction set, giving trivial examples on how to use most of the instructions available, and maybe an overview of the computer system (just enough to run some not-quite-as-trivial-but-still-useless-in-the-real-world examples (not all books are like this, but many are)). Also, some books assume you've never programmed a computer before. Other's assume some programming experience (BASIC, C, Fortran, Pascal, stuff like that) and try to use that knowledge to teach you. I've found that that usually is a hindrance to learning something so radically different (and no, C is not quite the 'portable Assembly' some say it is).

What I'm planning on doing is to present some (hopefully) realistic situations that face the Assembly Language programmer, offer several solutions, and explain why I did what I did. I also hope to teach something about the art of optimization (or, in the old days of hacking, what was called bumming code [1]). I also hope to teach you about what's available on the Amiga, both software wise (Exec, AmigaDOS, Intuition, etc) and hardware wise (DMA, Blitter, Copper, etc).

I've found that the best way to learn Assembly Language is to have a good book (or some book if you can't find a good book), look at as much Assembly code as you can get (the good, the bad and the ugly) and then program as much as possible (typing in examples by hand, if you have to).

So grab that 68k reference, the RKMs [6] and come on ...

* * * * *

But first! A word about the software I use (simply because Dan Abend (the almighty Editor-in-Chief) mentioned that I should mention it) and the hardware I have. So here goes ...

The Assembler I use is A68k [3] by Charlie Gibbs (well, the Amiga port anyway). The reason I use A68k is that one, it's free. Two, is that you get the source code. Also, it does support separate compilation of source code for later linking. It may not be as fast as ASMOne, but ASMOne doesn't really allow you to have separate modules (at least the versions I've seen).

The linker is Blink [3], from the Software Distillery. It too, is freely available and (from what I've heard) a bit better than Alink (the default linker for the Amiga)). Besides, A68k works in conjunction with Blink anyway, so ...

And the editor I use is Dme [3], by Matt Dillion. If you've never heard of Matt Dillion, that's surprising, because Matt is a code factory (Dme, Dmake, DICE (an integrated C environment including front end, pre-processor, compiler and assembler (not sure about the linker though)), Csh (a replacement for the CLI) and various other miscellaneous utilities). The editor is, again, free, comes with source code and is customizable (remap the entire keyboard if you want to).

About the only thing that isn't quite freely available are the include files from Commodore. You basically get them from Commodore, or from a development environment like Lattice C (that's where I got mine from), or type them in from the RMK books (not something I recommend 8-). I use them because they're there, but if many of you don't have them, I guess I can supply the values (mostly offsets inside of structures really) needed.

As for hardware, I have an Amiga 500 with a lowly 68000 running 1.3, 3 meg RAM (512K of that is chip) and a GVP Impack Series II A500-HD+ 50M hard drive (at least, that's what it says), and two 3.5 disk drives. Nothing that spectacular (well, except for the hard drive), although for development work, a harddrive is recommended, but you may get by without one (which I've done on a old IBM PC. Not for the faint of heart).

But enough of formalities, on with the show ...

* * * * *

Seeing how this is the first article (of a series) and all, I've decided that a frivolous program wouldn't be all that bad. Besides, the following program is one that is done in each programming language (at least, every time I learned a new language, I've done this program in it), and if you can get this one to run, you're on your way. Besides, even K&R [4] did this program in their book, *The C Programing Language*.

All the program does is print 'hello world' to the screen (well, in this case, to the CLI window you run the program from), but it does illustrate some points about the structure of an Assembly language program, and some of what is needed to start a program on the Amiga.

The first few lines include the Amiga specific include files. I've found that this order of includes works (at least for the Assembly .i files from SAS/Lattice and A68k). I've included them because (1) they're there, and (2) I don't feel like tracking down the offsets manually through the RKMs (although, like I said, I can probably find out the offsets they provide if enough people ask me ...).

Amiga Assembly References

[1]: This, and much more about the origins of hacking systems can be found in *Hacker's Heroes of the Computer Revolution* by Steven Levy (ISBN: 0-440-13405-6 [2]). Good book, and I recommend it.

[2]: ISBN stands for (I think) International Standard Book Number. Each book has one, and is all you need to order a book from a book store. The salesperson may say they need the title and the author for ordering information, but they're lying.

[3]: The following programs are available from the Fred Fish Collection.

Program	Fish Disk	Version
A68k	FF521	(2.71)
Blink	FF040	(6.5)
Dme	FF530	(1.45)

[4]: K&R Stand for Brian Kernighan and Dennis M. Ritchie, creators of the C programing language and that operating system known as UN*X [5].

[5]: The *New Hacker's Dictionary* (ISBN: 0-262-18145-2 (hc) or ISBN: 0-262-68069-6 (pbk) also available in an on-line version) says about UN*X:

UN*X: n. Used to refer to the UNIX operating system (a trademark of AT&T) in writing, but avoiding the need for the ugly {(TM)} typography. Also used to refer to any or all varieties of Unixoid operating systems. Ironically, lawyers now say (1990) that the requirement for the TM-postfix has no legal force, but the asterisk usage is entrenched anyhow. It has been suggested that there may be a psychological connection to practice in certain religions (especially Judaism) in which the name of the deity is never written out in full, e.g., 'YHWH' or 'G--d' is used. See also {glob}.

[6]: RKM stands for Rom Kernel Manual and is a three (four for 2.0) volume set of reference manuals for the Amiga. For AmigaDOS Version 1.3, they include:

Libraries & Devices
(ISBN: 0-201-18187-8, Commodore Item Number (CIM) 363099-01)

This volume is more or less a tutorial on programming in the Amiga environment, and includes using libraries, devices, graphics and Intuition.

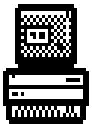
Includes & Autodocs
(ISBN: 0-201-18177-0 CIM: 327271-06)

This volume contains a description of all system calls available in the various libraries and devices. It also contains information about the IFF format (which I'm not going into), the source code for the C header files and Assembly include files as well as some sample code.

Hardware Reference
(ISBN: 0-201-18157-6 CIM: 327272-04)

This contains the hardware specification of the Amiga. No serious Amiga programer should be without it.

All are published by Addison-Wesley Publishing Company, Inc. I'm not affiliated with them, this is just for your information.



What My Amiga Means To Me

by Chuck Kenney

Radio controlled cars and airplanes, CB radios, police scanners, calculators, remote-controlled consumer electronics of all kinds, etc... I've bought them all - usually within days of their first appearances on the market! OKay-Okay, I admit it. I'm a hard-core neo-techno-phile. Or is it techno-neo-phile?

It was all the rage in the late seventies. Many of my fellow computer technicians were buying 'home computers' for themselves. "Home computer?", I thought. "I can have my own computer at home?". "I've gotta check this out!". So I drove down to my local RADIO SHACK. I took one look at this big grey thing that was called a 'home computer'. "It has a TV screen?", I asked. "Oh, it's called a 'C-R-T'. Sorry." "What does C-R-T mean?", I wondered. "And you can save programs on a cassette tape?". "Nice." <blah, blah..$$$.> "It costs HOW MUCH ???". For THAT kind of money, it wasn't quite nice enough! I mean, a home computer should be like a color TV with a keyboard - you know, it should do COLOR and draw PICTURES and make MUSIC! I decided to 'pass' on this T-R-S... whatever it was called. So I bought the latest and greatest printing calculator. "Never walk out of a technology store empty-handed." Oops! Batteries, extra printing paper... and oh....a pair of mini-book-shelf speakers while I'm here. ...And speaker wire.

Not more than a month later, I saw THE AD in a magazine... It looks like a keyboard with a computer built-in! It does color! It hooks up to your color TV!! It has special graphics and sound!!! And look at the key-caps on the keyboard!!! LOOK!! There's little graphics pictures right there on the fronts of the key-caps!!!! WOW!!!!!

My next income tax return was plowed into my new Commodore VIC-20. Three hundred and ninety-nine dollars... plus tax. I think that's what I paid. I was so excited the day I bought it, I can't remember now what store I bought my VIC-20 from. I only remember this colorful box under my arm and driving home to get started with my new 'home computer' hobby! How long ago was that? 10...15 years? It was exactly 4 apartments, 1 job, 1 wife, 1 home and 1 baby ago, all of which are still 'in service', even the apartments (presumably). During that time, and mostly before I got married, I had acquired a Commodore 64, Amiga-1000 and Amiga-2000, and let's not forget ALL the requisite accessories and peripherals; disk drives, hard drives, video, music, monitors, MIDI, memory, modems, aaannnnd multitudes of software. These are all still 'in service'. OKay-Okay, the VIC-20 is in a closet.

So, as I have grown/ aged/ matured/ withered (choose one) from a 'typical' American bachelor (who used to blow his entire paycheck) to a 'typical' home-owner/husband/father (who is saving for a hair-cut), I will now (finally) tell you what my Amiga (specifically) and Commodore computing (in general) has come to mean to me... or maybe you can guess...

I like to compose music - DMCS, SoundScape, Music-X, Dr. T's; Photography is another hobby Digi-View, ECA PhotoLab, TV*Text, DeluxePaint IV; Videos for friends and family - Deluxe Paint IV, Deluxe Video III, The Director, Animation station, etc; Pocket Billiards is another hobby - Animate the trick shots I know (BIG unfinished project!); Golf! I'll never break ninety - maybe if I digitize a pro's swing and genlock it over my own swing...; I probably SHOULD keep my resume updated - Final Copy and my wife's, and sisters', and in-laws'... - Final Copy again; Mailing lists for a couple of local music groups - SuperBase; Games - Too many to mention, but the C-64 "JUMPMAN" is a CLASSIC (I wrote a program to pick the start level!); What if I opened a Billiard Parlor? - VIP_Professional; Personal inventory - SuperBase; Interactive video for my daughter, Faith - AmigaVision; and on and on and on...

As my life habits and routines have changed over the years, so have my computing habits changed, from weekend-long cold-pizza-programming to intermittent tinkering. When I first purchased my Commodore 64, the first thing I did was to convert all my VIC-20 cartridge games to C-64 disks. This was a fun project for learning machine language, disk drives, memory maps, memory expanders, etc. My first Amiga project was to convert a C-64 "Simons's Basic" program I had written to AmigaBasic - the program simulates a Spiro-Graph (remember that set of plastic rings and wheels that you could draw neat little designs with?). Shortly before getting married, I purchased my first 'C compiler' and a couple 3-D ray-tracers. I managed to get the famous "Hello, world" C-program to compile, and to generate a glass sphere in TurboSilver, but since then I have become a devout home-owner/husband/father and the 'C' programming and ray-tracing doors will likely not ever be opened any wider by me. Well, maybe when I retire - it's only about 30 years away.

Only 30 years away? You may laugh...I'm serious! I met a retired 'home-computer nut' at an 'in-law' family reunion. He's a great-uncle of my wife. He was thrillfully telling me about his COCO (Radio Shack COlor COmputer) computing hobby. He had just hooked-up his brand new "5-1/2 inch disk drive". "5-1/4, Uncle COCO, not 5-1/2", I thought. He was amazed at how much faster it was than his data-cassette drive! So, I just MAY be sitting down in front of my old Amiga in the year two-thousand-and-twenty-something to try some ray-tracing as a retiree. I am quite sure I won't have the money to buy an Amiga 9000 or whatever Amiga model will (hopefully) be available by then. It WILL be fun though to bore a great-nephew-in-law computer hobbyist at a family reunion with a story of my recent upgrade to 8 Megs of memory! And he will probably be thinking something like, "God! Doesn't he know that RAM went out with the 1990's? NAM modules are the ONLY way to go these days.". (writer's note: NAM - Neural Access Memory module, infinite capacity, uses turn-of-the-21st-century neuralectronics technology.)

... But, I am getting off the subject...

So, at the risk of sounding ridiculously romantic, the Amiga has become a sort of "friend" - one I intend to keep for a long, long time. The Amiga has depths I will never explore. It has POWERFUL programs that can be intuitively learned (and revisited after long periods of time). "Ami" is both EASY to use AND to be PRODUCTIVE with - a rare combination. I get the feeling that the Amiga has the power to enable the prodigy, REGARDLESS of his field of interest. EVERY other computer I have EVER sat down to seems like only a box, a tube, and a keyboard - no personality, no excitement, no pizzazz. Just... well, not fun. Maybe this is why the other computers are making it in the business arena. After all, work IS supposed to be "work", you know... not "fun". Without the Amiga, I would have quite an un-interesting, un-fun, and difficult time getting various things accomplished.

Bravo, Amiga! Kudos, Commodore (not you, Marketing)!

... and you, dear reader, thanks for your time.

WHAT'S NEW continued...

This program is shareware and limited. In case you want the full version you can order CanonStudio from:

Wolf Faust
Am Dorfgarten 10
W-6000 Frankfurt 50
Germany
Tel: ++49-69-5486556 (GMT)
Fido: 2:249/3.5 (Wild Cat: ++49-6173-2544 HST,V.32bis)
Email: wfaust@aurea.hotb.sub.org

Or in England

JAM
75 Greatfields Drive
Uxbridge, UB8 3QN
Tel: 08952-74449 (GMT)

The price for registering is:

INTERNATIONAL US$25 includes shipping (+ $ 5 if payed by cheque. No COD!)

GERMANY DM 35 includes shipping. No COD (Nachnahme)!

ENGLAND #15 includes shipping

PS: Special Studio versions for Deskjet/Laserjet/24 pin Printers/9 pin Printers will soon appear, but more expensive... CanonStudio is an exclusive offer to Canon users and because of this, limited to Canon drivers.