

Amos

COLLABORATORS

	<i>TITLE :</i> Amos		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		November 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Amos	1
1.1	Amos.guide	1
1.2	the amos email list	1
1.3	hints and tips	2
1.4	system	2
1.5	processor type	2
1.6	amos pro compiler	2
1.7	graphics	5
1.8	amos 3d	5
1.9	om fileformat	5
1.10	non programing tips	5
1.11	amos pro config file	5
1.12	pal-ntsc	5
1.13	bugs	6
1.14	math	6
1.15	amal	6
1.16	(de)acceleration	6
1.17	the amos procedure library	8
1.18	the amos team	10
1.19	sfx site	10
1.20	bitplanes	11
1.21	varptr	13
1.22	menus	14
1.23	aminet	14

Chapter 1

Amos

1.1 Amos.guide

The Amos AmigaGuide help file main subjects:

The amos email list

The amos procedure library

The amos team

Sfx site

Amos 1.3 commands by BYTE BUILDERS INDUSTRIES (Alex Howell <↵
alexhl@nsq.nsq-gts.org>)

The Amos 1.3 editor by BYTE BUILDERS INDUSTRIES (Alex Howell <alexhl@nsq.nsq-gts. ↵
org>)

Amos Pro Compiler

Hints and tips

Bugs

1.2 the amos email list

Address: amos-list@access.digex.net

Purpose: For the AMOS programming language on Amiga computers. Features source, bug reports, and help from users around the world, but mainly European users. Most posts will be in English but there are no limitations to the language used, since AMOS is very popular in most European countries.

1.3 hints and tips

Programing subject:

Grapics

System

Amal

Non programing tips

1.4 system

Processor type

PAL-NTSC

Varptr

1.5 processor type

Date: 17 Jun 93 18:15:00 EST
From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
Subject: Re: Processor type
To: "amos-list" <amos-list@access.digex.net>

To find out the type of processor being used, look at the word at offset 296 in ExecBase [=Deek(Leek(4)+296)]. The bits are as follows:

Processor:

Bit 0 = 68010 or better
1 = 68020 or better
2 = 68030 or better
3 = 68040 or better

Co-processor:

Bit 4 = 68881 or better
Bit 5 = 68882 or better

So, to determine the actual processor, find the highest set bit in each field. That bit indicates the processor.

--Andy Church

1.6 amos pro compiler

Date: Mon, 12 Jul 93 15:46 GMT0BST-1
From: Phil South <snouty@cix.compulink.co.uk>

Subject: Amos Pro Compiler
To: amos-list@access.digex.net

This full review will be appearing in Amiga Computing. Or at least a version of it. Enjoy, my fellow AMOS fiends, or should that be friends? No I like fiends.

Phil

Review

AMOS Professional Compiler

Phil South reviews the latest addition to the AMOS family, but when all the dust settles is it any cop?

This is the AMOS product that's perhaps been more hotly debated and more eagerly awaited than just about any other. The thing is that AMOS as a development system was pretty much perfect, and the only thing missing from Pro, apart from a really good bug fixed version, was the compiler.

Now don't get me wrong, I know I'm always banging on about how Pro is buggy but that's not because it's completely no good. No no no. It's just that it's at the stage that AMOS was a few years ago when it first came out. It just needs a lot of use before all the bugs in the new features can be ironed out.

You see AMOS Pro is actually still the basic AMOS program you know and love, but there's been a lot of augmentation, and it'll take a while for the code to settle down after all the additions have been used a lot and all the bugs ironed out. But it will happen, and truth be told I'm going to start using Pro a lot more now. I know I said it would take a while to make me switch from AMOS Classic, and I know I'll still have that to fall back on should Pro let me down. But it's time to start having a little faith in the new program, and helping to iron out bugs rather than just sitting here griping about them

New Compiler Old Compiler

The same things I just said about the Pro compiler are true about Pro Compiler. It's still basically the same engine under the hood, just a new 1993 body and few technical enhancements which affect the way the thing drives, a bit like a Mondeo.

The face is different, that is to say that the front end program is much altered, and it runs from an icon rather than being loaded into AMOS, but I guess this just means that the program has been compiled. The compiler works with previous versions of the AMOS program like AMOS and Easy AMOS, so you can buy the new compiler even if you don't yet have the newest version of the language.

New Commands

The compiler obviously copes with all the new versions of AMOS Pro and all the new commands which have been added to the language since the last compiler was built. This obviously takes in all the ANIM player commands and music player commands too, and the library for

compiled programs to share means small programs if the target computer for your programs is running the correct amos.library. Even though you should use the shared library setting if you want to be sure people can run your programs, this is a big saving on size and so should be considered if you know everyone who'll be using your code.

You can do the usual squash command to compress the files, but you know every time I use this something goes wrong, not just in Compiler Pro but in the old Compiler too. I tend to forget that and use PowerPacker for online compression and decompression. The remove default screen problem is still there too. If your program uses and unpack command then the screen is automatically created into your default screen right? But if there is no default screen the screen can't be created, so no go.

You can also load in ASCII files for compilation, which is handy if you want to develop your programs on a laptop or palmtop word processor and transfer them for compilation. (I could use my trusty Portfolio, even!) There is talk too of this compiler being even faster than the original compiler. That is very hard to test, and really I couldn't see any difference. (If anyone at Europress can send me a reliable benchmark program, I'll be happy to run it, if not print it in the next issue!) But the program sizes were very different with a shared library. 61K including library plays 18k without. Not bad, but not totally portable to any machine which doesn't run the amos.library, so not really as good as a fully optimised compiler which would not include any commands not used in the program itself.

Free AMOS Pro 2.0

The program comes free with an updater to make your you have the latest version of AMOS Pro on your machine. I must admit I found this a little bit odd. I installed the AMOS Pro program about 10 seconds before I tried to use the updater, and I got an error message saying I needed to use the correct path. I was using the correct path, dammit, and I couldn't get it to change it's tune, so I killed it and installed to floppies, then RE-INSTALLED the new version to hard disk. This got up my nose more than a little, but I still can't figure out what the problem was.

Conclusion

Despite all these niggles and quibbles I do think the release of this program should be applauded, for no other reason than it allows us serious AMOS users to put our backs into it and see if AMOS Pro really has what it takes to be the Amiga development system of the 90s.

Obviously seasoned users will see that this is not anything new, but rather an upgrade of what has gone before. It's a tool, it does the job, but as my old man would say "if I wanted something to write home about I'd have been to the circus." Funny chap, my old man.

(ENDS)

1.7 graphics

Amos 3d

Bitplanes

1.8 amos 3d

Om fileformat

1.9 om fileformat

The om fileformat can be found in the file SimStructs on the object modeller disk

1.10 non programing tips

Amos pro Config file

1.11 amos pro config file

Date: 7 Sep 93 14:32:00 EST
From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
Subject: Re: Configuration
To: "amos-list" <amos-list@access.digex.net>

>What is the name, and location of the configuration file used by AMOS
>Pro? I added an extension, it won't run now, etc. I'm sure there are
>lots of people out there who have had this same problem.

It's S:AMOSPro_Interpreter_Config.

--Andy Church

1.12 pal-ntsc

Date: 12 Sep 93 11:41:00 EST
From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
Subject: Re: PAL/NTSC
To: "amos-list" <amos-list@access.digex.net>

>Is there an OS call to change between NTSC and PAL? Or would it be easier
>to Poke something? It can't be too hard, Mode_Switch is only 4K... can
>someone help?

Just Doke \$DFF1DC,x. For NTSC, x=0; for PAL, x=\$20.

Note that this only works on ECS systems (and probably AGA, although I haven't been able to test it).

--Andy Church

1.13 bugs

Math

menus

1.14 math

Date: 15 Sep 93 12:08:00 EST
From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
Subject: Re: Amos bugs, and thankyous
To: "amos-list" <amos-list@access.digex.net>

> Another bug I found is with the arccos function (hold on...but you don't NEED
>arccos for vectors.....well, tis fixed now but still). When you feed in very
>small values (order of about $1 \cdot 10^{-7}$) it returns ludicrous values. tsk tsk!

It seems that a lot of the transcendental functions give bad return values in certain cases. Try Cos(90) and see what you get. It's not 0, but something like 10^{-7} , which was enough to make my vertical lines diagonal.

[18-line signature deleted]

Please, don't get carried away! Signatures are fine, but not HUGE ones!

--Andy Church

1.15 amal

(de)acceleration

1.16 (de)acceleration

From: Mikko Makela <mmmakela@cc.helsinki.fi>
Message-Id: <199309151748.AA05528@kruuna.helsinki.fi>
Subject: AMAL-routine
To: amos-list@access.digex.net
Date: Wed, 15 Sep 1993 20:48:22 +0300 (EET DST)

Quite many of you showed interest with this, so I'm sending

this to the list (it's < 2k).

I decided to make little demonstration program out of it, just to make it easier to understand.

-----cut here-----

```
'This is a small demo program to show, how to use AMAL's Move-command
'for (de)acceleration of objects.

'Programmed by Mikko Mdkeld

BALL=4 : Rem amount of balls
_TOPSPEED=200 : LOWEST_POSSIBLE_STEPCOUNT=4
'R2 must not > RB (topspeed)
'RC will tell the lowest possible number of steps in movement.
Amreg(1)=_TOPSPEED : Amreg(2)=LOWEST_POSSIBLE_STEPCOUNT

'if ball hits the wall, main program will set R4=1, so
'that the Move command will be initiliazied.
A$="AUtotest (If R4=1 Jump Wall "
'if ball must be stopped, R3 will be set to 1, and
'to continue the movement R3->2
A$=A$+"If R3=1 Jump Hold ;If R3=2 Jump Go on "
'R2 will be decreased every time the autotest is run, and
'if it's <1, ball will be stopped.
'R2 is also used to count in how many steps will the movement be done.
A$=A$+"If R2<1 Direct Owait else Let R2=R2-1 eXit "
A$=A$+"Wall;Let R4=0 Direct Begin "
A$=A$+"Hold;Direct Sleep; Go on;Let R3=0 Direct Begin)"
'Here is the Move command
A$=A$+"Begin;If R2<1 Jump Owait; Move R0,R1,RC*RB/R2; Jump Begin "
A$=A$+"Sleep;Wait ;Owait: If R2>0 J Begin Pause J Owait "

'let's make an boject to move
Circle 100,100,7 : Paint 100,100 : Get Bob 0,1,90,90 To 111,111 : Cls 1

'init each ball
For L=0 To BALL-1
  Channel L To Bob L
  Bob L,80+L*15,80+L*15,1
  Amal L,A$
  Amreg(L,4)=1
Next L
'try different values for speed,range & angle
RSPEED=5000
RANGE=50
ANGLE#=3.3
For L=0 To BALL-1
  Amreg(L,0)=RANGE*Cos(ANGLE#-L*0.1) : Rem x-movemnt
  Amreg(L,1)=RANGE*Sin(ANGLE#-L*0.1) : Rem y-movement
  Amreg(L,2)=RSPEED/RANGE+1 : Rem speed
  Amreg(L,3)=0 : Rem just to make sure
  Amreg(L,4)=1 : Rem to init movementsequence
Next L

Synchro Off : Amal On
Repeat
```

```

For L=0 To BALL-1
  'test for left screenborder
  If X Bob(L)<1
    Amreg(L,0)--Amreg(L,0)
    Amreg(L,4)=1
    Bob L,0,,
  End If
Next L
'try without wait!
Wait 2
Synchro
Until Amreg(0,2)<1
End

```

1.17 the amos procedure library

The AMOS Procedure Library is a library of procedures for use in ←
 AMOS programs. (Surprise, surprise... :-)) The procedures were submitted to me by various people on the AMOS list. They can be easily merged into your AMOS programs to save you the time of writing them yourself. The latest version of the Procedure Library can be found on AmiNet as dev/amos/ProcLib20.lha .

I accept any kind of procedure, large or small. However, Paul Hickman came up with the concept of Screen Eaters at one point - routines which clear the screen in interesting ways. I was hoping to see some more of those (hint, hint), and possibly their counterparts, Screen Vomiters.

```

Date: 1 Oct 93 12:06:00 EST
From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
Subject: Procedure Library v1.0
To: "amos-list" <amos-list@access.digex.net>

```

I am in the process of uploading the initial release of the AMOS Procedure Library to

```

Aminet
. I've run into some problems, but I expect the library to
be available in 2-3 days. The file is:

```

```

/pub/
    aminet
    /dev/amos/ProcLib.lha

```

Below is the list of procedures in the library. (ver 1.0)
 A recent list should be sent to
 The amos email list
 as soon as it is ready.

--Andy Church

```

-----
_DOSERROR[ERR]
  Returns the AmigaDOS error message for the given error code.
_FAST_ELLIPSE[XC,YC,RX,RY,_FILLED]

```

Draws an ellipse using integer math.
 _FAST_LINE[X1,Y1,X2,Y2]
 Draws a line using integer math.
 _MAKEMASK[_SCREEN,_COLOUR,TEMP1,TEMP2,X1,Y1,X2,Y2,_ICON]
 Makes a mask for an icon with a given transparent colour.
 OPENFONT[NAME\$,SIZE,_REM]
 Opens or closes a font.
 _PALETTE_TO_ANTIQUA[_SCREEN]
 Converts palette to "antique photograph" colours.
 _PALETTE_TO_BW[_SCREEN]
 Converts palette to black and white.
 _PALETTE_TO_NEGATIVE[_SCREEN]
 Converts palette to negative colours.
 _SEPARATE_RGB[_SCREEN,X]
 Filters colours in picture.
 _SHADOW_TXT[X,Y,_FG,_BG,_TXT\$]
 Writes shadowed text to the screen.
 _TO_NTSC
 Switches display to NTSC mode.
 _TO_PAL
 Switches display to PAL mode.

Date: 21 Sep 93 17:40:00 EST
 From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
 Subject: Procedure library (again)
 To: "amos-list" <amos-list@access.digex.net>

When you send me a procedure for inclusion in the library, please also include a short "readme" file (like those on Aminet) that describes what the procedure's function is. If possible, I'd also like an example program that I can use to test the procedure, just to make sure it works.

I'd prefer that you use the following format for your readme file:

[Required items]

NAME_OF_PROCEDURE

Description : A short description of the procedure's function. 1 or 2 sentences should be enough in most cases.

Calling format: How to call the procedure, and what its parameters are. E.g. MYPROC[num1,num2,sumadr]
 num1,num2 - numbers to work with
 sumadr - ADDRESS of sum (use Varptr())

Return value : What your procedure returns. If none, write "None".

Author : Your name, and your e-mail address.

[Optional item]

Special notes : If your procedure only works with AMOS Pro, or if it requires a special library, or another procedure, say so here. If there's anything else you think I should know, put that here too.

--Andy Church

1.18 the amos team

team@slick.clare.tased.edu.au

1.19 sfx site

Date: 22 Sep 1993 17:36:23 -0500 (EST)
From: Tom Plackowski <tom@elms.concept.com.au>
Subject: Re: SoundFX

>Hello, I need some SFX (fairley small if possible) for my game.
>Some good explosions, missile and changun sounds, and the sound of
>a helicopters rotor.

>Does anyone have or know where I can get sounds like these?

Have a poke around at:

SOUNDS.SDSU.EDU

This is their readme:

This is the brand new (well, almost...) machine which will provide access to the sun sounds archive.

All sounds are in /sounds.

ATTENTION FRIENDLY HELPFUL ARCHIVE USERS:

I am in need of some help. As you will notice, each directory now contains a file called 'INDEX' which is a list of the sounds in that directory with an optional description. This discription is to be gotten from the header of the sound file. My request is: Most sounds do not have an sort of info with them. Could you be so kind to send me some descriptions of sounds so that I can add them to the headers? This would really help out me and other people as well.

The INDEX files are generated nightly by a program I wrote called soundinfo This program is also available for anonymous ftp in /pub/soundinfo.shar It also allows one to change the header info string of a sound.

If you have a request for a certain song, please don't mail the request to me since I am not very likely to have the song on CD. However, you put requests in the /sound_requests directory. If everyone with record abilities will look in that directory every once in a while and record some of the requested sounds, it would make a lot of people a lot happier. (If you do upload a sound that was requested, please remove the original request, so multiple people do not upload the same sound...)

If you want to know how to play these sounds, how to convert them, or how to annoy other people with them, please read the README.sounds file in this directory.

As always keep those sounds coming!

Always upload new sounds in /incoming, please.

I would very much like to expand the database of sound effects, cartoon

sounds and things like that, so if you have some, upload them!

If you have questions/comments/money, send mail to turtle@sciences.sdsu.edu

--Andrew

Andrew "Race Turtle" Scherpbier, SysMgr
College of Sciences
San Diego State University
turtle@sciences.sdsu.edu
Voice: (619) 594-5026
Fax: (619) 594-6381

// Tom

1.20 bitplanes

From: sfmcnally@BIX.com
Subject: Bitplanes
To: amos-list@access.digex.net

Well, bitplanes are how all graphics on the Amiga are stored.
(If you understand Binary numbers it's quite simple)

Every image is made up of a number of bitplanes according to how many colors it uses, 1 bitplane for 2 colors, 2 planes for 4 colors, 3 planes for 8 colors, and 5 and 6 planes for 32 and 64 colors. (You can also go to 7 and 8 planes for 128 and 256 colors on AGA machines)

In every bitplane, one bit represents one pixel, so a byte (8 bits) of a bitplane would be 8 pixels across.
At it's simplest, with one bit plane wherever there's a set bit in the bitplane that pixel will be in color 1, a binary 0 in one of the bits means color 0 at that pixel.

To get more colors the system adds up the bits of all the planes at a certian pixel and that makes the color number.
Say at pixel A plane 1 is set to 0, plane 2 is set to 1, and plane 3 is set to 1, combining those will make the binary number %011 which is 6 in normal numbers, so at that pixel there would be color 6.

Now to change the colors of bobs you can tell the blitter NOT to draw certian planes, so if in the above example you told the blitter NOT to draw plane 2 the color would be %001, or 4.

Now for a turtle on a 16 color screen, you should draw the turtle images in color 15, 15 is %1111 in binary, so then you can change the turtle to ANY color JUST by not drawing certian combinations of planes on the screen! :)

Lets see... ::going for manual::

Ok, if you have a bob drawn in color 15, on a 16 color screen, this bit of code will draw the bob in color "COLOR":

```
Set Bob 1,,COLOR,
Bob 1,X,Y,TURTLE_IMAGE
```

Where turtle_image is the image number for the turtle.

:)

Seumas

```
Date: 27 Sep 93 21:23:00 EST
From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
Subject: Re: Bitplanes
To: "amos-list" <amos-list@access.digex.net>
```

A little clarification here...

```
>In every bitplane, one bit represents one pixel, so a byte (8 bits)
>of a bitplane would be 8 pixels across.
>At it's simplest, with one bit plane wherever there's a set bit
>in the bitplane that pixel will be in color 1, a binary 0 in
>one of the bits means color 0 at that pixel.
```

This is right. However...

```
>To get more colors the system adds up the bits of all the planes
>at a certian pixel and that makes the color number.
>Say at pixel A plane 1 is set to 0, plane 2 is set to 1, and plane
>3 is set to 1, combining those will make the binary number %011
>which is 6 in normal numbers, so at that pixel there would be
>color 6.
```

This is wrong -- you're numbering the planes backwards. What you should have said is this:

```
Say at pixel A plane *2* is set to 0, plane *1* is set to 1, and plane
*0* is set to 1, combining those...
```

The plane number is the same as the bit number in the colour. So plane 0 corresponds to bit 0 (the least significant bit), plane 1 corresponds to bit 1, etc.

You also don't know as much about binary as you seem to think you know. The binary number 011 is equal to 3 in decimal ($0*2^2 + 1*2^1 + 1*2^0 = 0+2+1 = 3$).

```
>Now to change the colors of bobs you can tell the blitter NOT
>to draw certian planes, so if in the above example you told the
>blitter NOT to draw plane 2 the color would be %001, or 4.
```

^^^^ NO!!! 1, not 4!!!

You'd tell it not to draw plane 1 in order to get %001 from the above case.

That is, you'd use "Set Bob N,,%11111101," where N is your bob number.

--Andy Church

1.21 varptr

Date: 27 Sep 93 21:34:00 EST
 From: "Andrew Church" <95ACHURCH@vax.mbhs.edu>
 Subject: Re: Use of Varptr
 To: "amos-list" <amos-list@access.digex.net>

> Last night I was stuffing around with passing Varptr's as parameters to
 > procedures, trying to pass the address of an array of floating point numbers.
 > The problem is that I couldn't just Poke(Varptr(ARRAY)+OFFSET) to get to
 > ARRAY(OFFSET). I noticed that each successive Varptr was 4 units apart, and
 > this morning I realized why the values I Poked weren't what was stored.
 > So could someone tell me

>* How to use Varptr, Poke and Peek to store (properly) and read from an array
 > of integers?

Writing: Loke Varptr(ARRAY(0))+ELEMENT*4,_VALUE
 Reading: _VALUE=Leek(Varptr(ARRAY(0))+ELEMENT*4)

where ELEMENT is the element number you're using, just like ARRAY(ELEMENT).

Of course, this is only useful if you've passed the array address, as in

```
_PROCEDURE[Varptr(ARRAY(0))]
```

since otherwise you can just use _VALUE=ARRAY(ELEMENT).

>* How to use Varptr, Poke and Peek to store (properly) and read from an array
 > of floats? (Ie, the set up of the float, such as the mantissa length...that
 > sort of garbage, and where they are stored wrt the array's adress).

They're also stored as 4-byte values, so use the same scheme I described
 above to access them.

The format depends on whether you're using Creator or Pro. Creator uses FFP:

```
31                876      0  
| - - - mantissa - - - |S| exp |
```

where m is the mantissa (with an assumed binary point after the first digit),
 S is the sign bit and exp is the bias-64 exponent. In other words,
 exp = exponent+64

Pro, on the other hand, uses IEEE single-precision. I'm not entirely sure
 about the format for IEEE floats. If you use Set Double Precision, that
 switches Pro to 8-byte double-precision floats, which is something completely
 different.

--Andy Church

1.22 menus

Subject: Menus & multitasking
 To: amos-list@access.digex.net
 Date: Tue, 5 Oct 1993 10:44:22 +0200 (EET)

I think I found bug in AMOS menus.
 If I try something like this:

```
Screen Close 0
BEGIN:
Screen Open 0,320,200,16,Lowres
Menu$(1)="First menu"
Menu On
Wait Key
Menu Off
Screen Close 0
Screen Open 1,320,200,16,Lowres
Print "This is screen 1"
Wait Key
Screen Close 1
Goto BEGIN
```

...and while the program is running, go to WB, and close a window or screen, the AMOS program will give "Screen not opened"-error.
 Is this a bug, or is there something I should know about using menus with temporary screens ?

Mikko

1.23 aminet

Aminet is a huge collection of amiga pd stuff, it is mirrored to a lot places here is a list:

Country	Site	Directory
SCANDINAVIA	ftp.luth.se	pub/aminet/
SWITZERLAND	amiga.physik.unizh.ch	pub/aminet/
SWITZERLAND	litamiga.epfl.ch	pub/aminet/
GERMANY	ftp.uni-kl.de	pub/aminet/
GERMANY	ftp.cs.tu-berlin.de	pub/aminet/
GERMANY	ftp.th-darmstadt.de	pub/aminet/
USA	ftp.etsu.edu	pub/aminet/
USA	ftp.wustl.edu	pub/aminet/
USA	oes.orst.edu	pub/aminet/