Welcome to The Newsletter

Merry Christmas and a Happy New Year to all of you out there reading this amazing piece of modern literature! That in years to come will only be spoken of in hushed and reverent tones, well concealed away from the general population.

Well did Santa leave any AMOS presents in your disk drive this Christmas? Like a Compiler? No? You to Huh. Well thats ok, because if you did, I would have been around your place like a shot so that I could get a glimpse of it working. The Compiler is still a way off yet, Francois and Daisy are hard at it trying to get the compiler finished for a March release. Yes, we know that the last probable release date was February, but its pointlessly completing and releasing a Compiler if you have to go back and fix new bugs in the original language and then sit down and re-write the compiler. Its not an easy job writing a 100% bug free language! Turn to Daisys regular column for an informative update on the Compilers Features and Progress.

Well, this is the 5th Newsletter to come off the presses, and believe me, they seem to take longer and longer to write. (No, I'm not bored, there's just so much to put in each one!) Quite a few of you have said that you never received the latest Newsletter or it was really late. Well we can only apologize for Australia Post for the really efficient job they do of delivering our mail the next day?? As of the last Newsletter, hopefully Francois will be able to work this problem out...

Well what are you doing reading my ramblings? Turn the page and read on! Theres so much in here!

Newsletter Vol 5. Contents

PAGE 1...Welcome to The Newsletter
PAGE 1...Contents
PAGE 2...Beginners Tutorial
PAGE 2...Bugs/Typos/Errors/Red Faces
PAGE 2...The AMOS Hotline
PAGE 3...AMOS RPG & Adventure Club
PAGE 3...Ten Liner Programs
PAGE 3...Aussie Demo Competition Update
PAGE 3...The AMOS BBS Scene
PAGE 4...Unexpanded AMIGA Blues
PAGE 4...Graphics Mini Tips
PAGE 5...20 Things About Francois Lionet
PAGE 5...Submitting Articles
PAGE 5...Letters To The Editor
PAGE 5...AMOS In Education
PAGE 6...Daisy's Regular Column
PAGE 7...AMOS File Selectors - A Tutorial
PAGE 7...Modifying Your Sprite Editor Even More
PAGE 8...PD Update
PAGE 8...AUSSIE Disk Listing
PAGE 9...BRITISH PD Listing
BEGINNERS Page

BEGINNERS Tutorial

L L L

L

Leamer programmers seem to be split into two groups: Those who have programmed in Basic before but not AMOS, and those who have never programmed at all.

For this reason I am going to split this series into two separate articles: ABSOLUTE BEGINNERS will help those who have never programmed before, and ALMOST BEGINNERS will cover aspects of programming a level up, for those just converting to AMOS or who have never attempted a major program before.

So for our Absolute Beginners, in this issue we are going to write a very simple dice game. This program will introduce some very basic features of programming along with a few handy tips.

There will be a one small programming convention here: Because AMOS can have lines longer than we can list on one line in this newsletter, I will be using line numbers for this program instead of labels so that you know when to enter a new line. We will convert to labels in a later issue.

The first question you should ask yourself before writing a program is What am I Doing?? A program is a sequence of commands that the computer follows exactly, so you must make sure you get them exactly right! By deciding beforehand what you are going to do, you will make life a lot easier.

For our dice game, the rules are simple: The computer rolls a dice to produce a number between 1 and 6. Then the player rolls a dice for another number between 1 and 6 and the highest roll wins. We also want to add a few embellishments such as the ability to enter the player's name, and to play again.

Before writing the program, it can be beneficial to work it out in PSEUDO-CODE. This is when you write down a summary of what the program will have to do, without having to remember all the commands.

Our program will need routines to:
1) Enter the players name.
2) Roll the computer dice and display it.
3) Roll the dice and display it. (With the players name)
4) Check for a winner.
5) Ask player if he or she wants another go.
6) If so, then go back to step 2.
7) Otherwise exit.

And that's the game in a nutshell. As you can see, Pseudo-code makes the program look less complicated. Believe it or not, it really is that simple.

The first task on our list is to enter the player's name. We will have to store it somewhere as we will need it later on to display with his dice roll. So a STRING VARIABLE must be used. As the name is made up of a string of letters, a STRING VARIABLE has to be used (Identified by the $ after the Variables name). We will call it NAMES.

The command required to get an input value from the keyboard and store it into a variable is INPUT. The type of value depends on the variables being input for input - for instance you can't input strings into a numeric variable. You can also add a prompt to an INPUT command so you can tell the user what information is required. So our first line will be:

10 INPUT "Enter your Name Please and Press RETURN"; NAMES

This will display the message: Enter your Name Please and Press RETURN$, then wait for the players name and on pressing return, whatever was typed will be stored in the variable NAMES.

Next we need to generate some random numbers for the dice throws. As we are only dealing with the numbers 1 to 6, we can use normal INTEGER VARIABLES in which to store them.

So that the numbers we generate will be different every game, we need to add a command which SEEDS the Random Number Generator. (Otherwise the numbers would be the same every time!) So we add the line:

15 RANDOMIZE TIMER

The function RND(n) returns a value from 0 to n, so to get a number from 1 to 6, we simply use variable=RND(6)+1. We don't use RND(6) because we don't want the number 0 generated for any throw of the dice.

We will call the variable for the computer's throw D1 and the players throw D2, so the very next lines will read:

20 D1=RND(6)+1
30 D2=RND(6)+1

To display the scores, we simply use the PRINT statement, which can output text, numbers or a mixture of both. First we display the computer's score:

40 PRINT "The Computer Rolls"$; A$; D1

Then we need to display what the player rolled:

50 PRINT NAMES$; Rolls A$; D2

Notice that on the player's display, we used the variable NAMES as the first part of the print which already holds his/her name. This means that the name will be output as part of the message.

The only tricky part of the program is working out who wins. To do this, we need to use an IF..., THEN statement to work out the argument given to it. If the result is TRUE, it will execute the series of commands after the THEN on the same line. If the result of the condition is FALSE it will go on to the next line. (Or whatever is after the optional ELSE statement).

First; we will check to see if the computer rolled the highest score. This is TRUE when D1 is greater than D2. This can be checked by the line:

60 IF D1>D2 THEN Print "The Computer Wins!"

Then we check to see if the player has the highest score:

70 IF D1<D2 THEN Print NAMES$; " Wins!"

And finally we check for a draw with the line:

80 IF D1=D2 THEN Print "It was a draw"

The meanings of the mathematical symbols in lines 60, 70 and 80 are:

> Is Greater Than
< Is Less Than
= Is Equal To

You can even have combinations such as >=, which means is Greater than or Equal to.

At we need to do now is see if the player wants another go. We can use another INPUT statement here and then check to see if the variable used contains YES:

90 INPUT "To Play Again Type YES and Press RETURN"$; Y$

Notice on line 100 how I've used the IF... THEN statement with a string variable and also a GOTO the line specified - in this case 15 - providing the result is TRUE.

If you've been typing in the program while reading this, then now hit F1 to RUN the program. (Running the program tells the computer to start executing the program). Congratulations! You have just written your first game.

Next Newsletter-More for ABSOLUTE BEGINNERS!

BUGS!

Yes thats right its embarassment time, Bug Fixes! Well so far we have not done to badly, no one reported any in Newsletters 1, 2 or 3 but in number 4 - Oh Dear! A number of people have rung up and said that my Rolling Mouse Menu Program had a bug! But before I go out and hit myself over the head with an Atari S1, I will say this in my defense...I was just testing you to see if you DO read every word of the newsletters! Well thats my excuse and I'm sticking to it!

In fact what I had neglected to do was to Include a Reserve Zone command before I called my procedural So all you have to do is work out how may menu items there will be and insert a line which says Reserve Zone X. With X being the number of Menu Items. Then it will work fine! I swear it will this time!

Watch this column for any other BUG reports in the future! But don't bother because there won't be any! :)

The AMOS HOTLINE

It would seem that in the last 4 issues of the Newsletter I have forgotten to include the AUSSIE AMOS HOTLINE-HELPLINE PHONE NUMBER! Well in this issue I haven't. But it could have been worse, this could be issue 401 and we might have only just remembered to put it in! You can ring this number during normal business hours for help with AMOS. But please keep a few things in mind when you do...

(02)748 4884

We are really only just like you-Learning AMOS. We might not know the answer to your question straight away, but we will certainly try and get you an answer as quick as we can. Although we hate to admit it, we are only human and sometimes we forget. But remember we will try and help you as quickly as we can.
The first is Weed War, which is a version of the old classic Missile Command. You must control your hover ship and try and stop the endless offensive puts from coming down and destroying your Groovy Garden. You can fire your weed killer by pressing the left mouse button, but remember you have a limited number of shots per level. If you get really desperate, you can fire your shield by pressing the right mouse button. (3 per level) You control your little ship with the mouse. The weeds will explode upon touching your weed killer, weed shield or your Groovy Grass. The game is over when the weeds manage to get through your Groovy Grass and into the ground! Here is the listing below.

```
1) Dim MX(50),MY(50) : Dir$='extras:Sprite_Sprites_500/aliens' : Load
'aliens.skr' : Get Sprites Palette : C$='LVL' : A$='AU' (TRG < X> XM JU' 

A$='AU'1 R1 Y1 W1 : Amal 1,A$ : Amal 0 : Autoback 0 : Curs Off : Flash Off : 

2) Screen 1,320,256,16,Lowres : Curs Off : Flash Off : Hide On : 

3) Palette 1,$B90,$AAO,$9B0,$8CO 3$7D0,$6E0,$5F0 

4) For O#=0.0 To SCRY# : For P#=0.0 To SCRX# : M#=XMIN#+P#'H# 

5) 800 : SCRY#=192 : X0 = MX(A) : Y0 = MY(A) : X# = MX(A)+20 : Y# = MY(A)+20 : 

6) Draw : Ink 0 : Bar 0,0 To 320,200 : Ink 4 : For A=1 To 10 Step 2 : 

7) U# = Ink 13 : A%=0 : Ink 4 : Bar 0,120 To 320,125+SHIELD*4 

8) If Mouse Key=2 and MKO<>2 and SHIELD>0 Then Dec SHIELD 

9) If Mouse Key=2 and MKO<>2 and SHIELD>0 Then Dec SHIELD 

10) Draw : Ink 0 : Bar 0,0 To 320,200 : Ink 4 : For A=1 To 10 Step 2 : 

11) END 
```

Ten Line Programming

10 Lines?! 

10 lines games, What are they? Where do they come from? What do they want? Why do we want them? 

10 line games are nothing more than a 10 line AMOS program. Well what so special about that you may ask? Well think about how hard it would be to sit down and write an entire game in just 10 lines! Not easy but it can be done! Look at the two examples in this article, they use 10 lines or less and are quite good for the size of the code. Why would you write a 10 liner? Well, 

1) They take very little time to code 

2) They don't take very long to type in 

3) They encourage efficient programming rather than neat but wasteful full programs 

4) Just for the hell of it!

Trying to cram an entire game into just 10 lines can be a major challenge! (Even for De潦y!) It is a great way of testing your programming skills as it forces you to write with maximum efficiency. Because, the more efficient you make your code, the more you can add to the game itself.

Just think, years ago programmers filled entire machines with programs so that we could play Space Invaders or Missile Command. If these programmers could see what we can do with just 10 lines now, they would retell

What we would like to see is who can write the best 10 liner game/program. (Preferably a game!) Each month we will publish some of the best submitted and the winning entry will be put into the library when we have about 50 or so. But first just a couple of simple rules. No AMAL Based (AMAL things are allowed) Sprites/Bobs used must be from the Sprites 500 collection. They must not be more than 10 lines long.

The current challenge is to convert Space Invaders, Asteroids, Xonix II and Shadowcasters I to 10 liner games! Can you do it? (Yes we know, Astaticos would be the hardest to convert)

Aussie Demo Competition

It would seem that we made a mistake by not accurately specifying the closing date of the demo competition. Well, rectify the situation here is the closing date. (That's right we have put it back to give you more time)

CLOSING DATE:

20th February 1991

We have received plenty of entries but lets see some more in! Refer to the previous newsletter for the address and details/rules.
Unexpanded AMIGA Blues

If you are working with AMOS on an Unexpanded Amiga, you will no doubt be very happy to hear that we will be publishing a series of regular articles on how to get as much as possible out of just 512K. Yes I know that 8K might not sound like a lot of memory, but on average you can fit another 30 or so sprites in 8K! But remember, when modifying any of the AMOS subroutines, make sure you work with your Backups! NEVER WORK ON YOUR ORIGINAL MASTER DISKS!

<Removing The NICENESS Routines>

A) Load the Sprite Editor and open the sprite editor program under the menu of EDITOR OPTIONS. (Don't forget to save your changes!)

B) Now quit AMOS and load your Art Package and load the IFF picture in 8 colour Low resolution Mode. Now erase the Shadow software logo and the 4 bars which where there before and to the right of the GET Button. To erase them simply black out the picture. Now save the picture.

C) Re-boot AMOS. (After first switching off)

D) Now load your modified Sprite editor and goto Direct Mode. Type the following...

Load iff_"Menu.Buttons iff"
Space off to 6.0.32
Load iff_"Main iff"
Space off to 6.0.32

E) Now return to the Editor and re-save the Sprite Editor as SMALL_Editor.AMOS

You should have now recovered about 7-8K of memory. If you want to save even more then read on some more!

<FUTURE DIMENSIONS - (07) 208 5004>

<GRAPHICS MINI TIPS>
20 Things You Have Always Wanted To Know About
Francois Lionet!

But You Couldn't Because You Didn't Have His Phone Number

Francois lives in a town called Yerres, located 12 miles from Paris. He was born on the 6th of July 1963 in Maubeuge, a small town near Belgium.

He is 27 years old, married to his wife Carine and they have a dog called Daisyl (Suprise Suprise) Daisy!

His two brothers are also as talented as himself. One is a dentist and married, and the other is a cardiologist, single and living in Tahiti.

Francois’ first computer was a Superboard II from Ohio Scientific, purchased in 1981. He was one of the only few French owners so he had to write his own 6502 assembler in Basic, and then his own games. He even wrote a full Defender game in just 8K of memory.

Francois wrote two games for Firebird Software: Chicken Chase (Oric, CPC and C-64 - over 60,000 copies sold) and Olie ( CPC). He also wrote three commercial Oric games for a French software house and a musical puzzle called SERENADE for the C-64.

He wrote the CAPTAIN BLOOD conversions for the PC and C-64 - most of this work was done in just a month! (Fast isn’t he?)

Francois programs on almost any microcomputer. But this has its problems. The worst case is programming on an Intel 8088 (IBM PC) and a Motrola 68000 (AMIGA) - Intel syntax works from right to left, and Motorola syntax works from left to right!

STOS took 18 months to write using Kuma’s K-SEKA assembler. It was first released in France in the Spring of 1986, but due to the way it was marketed it simply didn’t take off.

Mandarin bought the rights for STOS, added extra accessories and streamlined the package, then relaunched the product.

STOS zooms into the Gallup chart to number one in September 1988 - wow!

Francois was then asked to write the compiler after it was obvious that STOS was a big hit. This time he uses the much faster DEVPAC 2 assembler by Hissot.

AMOS is targeted as Francois’ next project. This time though, it should be a simple matter of taking the ST source code and adapting it to the AMIGA, making improvements to the system as he goes. How wrong we were! Due to Francois’ brilliance and a lot of input from expectant users it took another 18 months solid work! Again using DEVPAC 2. We’re sure you’ll agree it was worth the wait!

Francois programs listening to loud music through his headphones. He had to do a flashing light to tell him the phone was ringing!

Francois’ idols are Jes (Starglider) San, Bill (Sinbad) Williams, Jeff (Tripatron) Minter and all the Captain Blood Team. He would really like to be like them when he grows up!

Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

His favorite film is "Brazil" and he enjoys the Monty Python Films.

Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

Once he had acclimatized to the AMIGA, Francois disowned his ST, Once an Atarian now a Commodore man. The only hate he has about the AMIGA is the floppy disk drive access time) (Don’t we all hate il)

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?

# He enjoys to program sprite and bob routines the most and hates hacking out sorting procedures and structure handling code.

# Francois describes himself as: Perserverant, happy and messy (But not in his programs though).

# His favorite film is "Brazil" and he enjoys the Monty Python Films.

# Francois likes: Ski-ing, Squash, Tennis, Playing the piano, Reading SF comics, Sailing boats, holidaying, sun-bathing and Girls!

# If asked, if you were to be re-incarnated what would you return as, Francois replied: As my dog Daisyl! She sleeps when she wants, eats and plays all day - what more could one ask for from life?
A FAX From France!
Daisy's Regular Column!

Nice to meet you again in this superb newsletter! Here we go for another page filled with dog's talking. I am really busy programming the compiler right now. I'm going to give you a little insight on how it will look...

AMOS compiler will come as an AMOS program: you'll have to load it as an accessory, and call it from the editor. If you do so, it will compile the program you are currently editing. Of course, you will be able to compile the compiler, and create a stand alone program, clickable under WorkBench.

The compiler will only accept TESTED programs, so it can go faster on compiling. Anyway, if the compiler detects an error, it will return you to the faulty line in the program. As you can imagine, the AMOS program is only the compiler's graphics shell: it calls a machine language program that does all the compiling job. Some infos on it:

- 1 pass compiler, (in fact I tried to do a 0.5 passes compiler to be faster, but it was compiling half of the program :-)
- Options to set source program origin: from memory (FAST) or disk
- Options to set object program destination: to memory (FAST again) or disk
- Options to tell the library position: in memory (FAST FAST FAST) or disk

Doing everything on my hard drive, an AMOS program is compiled in a matter of 4 seconds. S well, I am quite proud of myself! Francois as always says it could be faster. This guy does nothing and always criticizes!

You will also have the ability to create AMOS runnable programs: They will be smaller, and will be loaded under the editor like any other AMOS program. The only difference is that they will run 3-4 times faster! Imagine, you compile the program you are currently editing. Of course, you will be able to compile the faster on compiling. Anyway, if the compiler detects an error, you won't have anyway all instruction set at your disposal: you will have the main point, you'll have all others: BOB, SPRITES, SCREENS, TEXT etc

If I have time, I would also like to do a CLI program generator: the compiled program would not display a screen, but use the default CLI when you do a PRINT. Easy way of creating a new CLI command!

So, the AMOS compiler screen will rather look like the STG8 one, but more fancy, with nice animations during compilation. You'll have a couple of buttons like:

- SOURCE origin
- OBJECT destination
- OBJECT type (Workbench, amos, boot disk)
- OPTIONS menu (to set compiler at the hearts content)
- COMPILE button (the most important one)...

That's it, I am a little bit late in the programmation, but we still look for a MARCH release!

You can find in PD the new SERIAL extension for AMOS. You certainly know that opening through AmigaDos a serial port does not really work under AMIGA. You won't have anyway all instruction set at your disposal: you will have to forget about serial commands and use DEBUG and LOADSAVE, both serial instructions (but LOADSAVE, lack of disk instructions). But this is the main point, you will have all others: BOB, SPRITES, SCREENS, TEXT etc

As I might have told you before, (No you didn't ED) I want to make a special option to create a boot disk that grabs ALL available memory in the AMIGA. You won't have anyway all instruction set at your disposal: you will have

I have time, I would also like to do a CLI program generator: the compiled program would not display a screen, but use the default CLI when you do a PRINT. Easy way of creating a new CLI command!

So, the AMOS compiler screen will rather look like the STG8 one, but more fancy, with nice animations during compilation. You'll have a couple of buttons like:

- SOURCE origin
- OBJECT destination
- OBJECT type (Workbench, amos, boot disk)
- OPTIONS menu (to set compiler at the hearts content)
- COMPILE button (the most important one)...

That's it, I am a little bit late in the programmation, but we still look for a MARCH release!

You can find in PD the new SERIAL extension for AMOS. You certainly know that opening through AmigaDos a serial port does not really work under AMOS. AmigaDos and I do not like each other! So I did a small extension file that adds a new command to AMOS, to handle serial talking. Ex: Serial Send, Serial Input, Serial Speed etc...

I look forward to receiving programs from Australia: Neil told me he was going to send some to me. I have already seen some incredible things in the English PD Library.

When the Compiler is finished, a wave of compiled AMOS program will arrive on classic PD library, and everybody will be amazed by the quality of the program!

Talking about quality, here is the second Bone Demo (*): Bone Demo II, The Come Back (*). In order to run it, you must load a music bank in direct mode, choose one that has a lot of rhythm for best effects.

BONE DEMO II _ THE COMEBACK(*)

BY DAISY LIONET

SCREEN SETUP

Hide On
Screen Open 1,352,430,2 lowres
Curs Off: Flash Off : Cis 1
Screen Open 0,352,430,2 lowres
Curs Off: Flash Off : Cis 0
Screen Display 0.30,340

Dual Playfield 0.1
Screen Offset 0.352,64
Screen Offset 1.352,64
Colour 1.0 : Colour 5.0
Screen Open 2.320,2.0
Curs Off : Colour 1.0 F40

View
Music 1

* AMAL SETUP
Channel 0 To Screen Offset 0
Channel 1 To Screen Offset 1
A0S=A0$+" Let R3=1; Let R4=1; Let R5=3;"
A0S=A0$+" Loop: Let R2 = Vu(0); If R2<0 Jump X"
A0S=A0$+" If R0=0 Jump F"
A0S=A0$+" Let R0=0; Jump F"
A0S=A0$+" X: Let R0=R2;"
A0S=A0$+" Let R3=2*F*R3;"
A0S=A0$+" F: Let X=Vu(2)*Vu(3)*"
A0S=A0$+" Let R2=Vu(1); If R2<0 Jump Y"
A0S=A0$+" If R1=0 Jump G"
A0S=A0$+" Let R1=1-R1-R5; If R1>0 Jump G"
A0S=A0$+" Let R1=0; Jump G"
A0S=A0$+" Y: Let R1=R2;"
A0S=A0$+" Let R4=2*F*R4"
A0S=A0$+" G: Let Y=R4*R1+64;"
A0S=A0$+" Pause; Jump Loop"

Aimal 0.50S=Instr("Vu(0)"); Mid$(A0$)="Vu(2)"
Aimal 0.50S=Instr("Vu(1)"); Mid$(A0$)="Vu(3)"

"DRAW CIRCLES + THE BONES"

Screen 0
BONE[176,200,40,10,15]
For C=1 To 256 Step 2
Circle 176,200,C

Next
Screen Copy 0 To 1

LET'S GO

On Screen Close 2

COLOUR CHANGES

Do
Read C1,C2
If C1=1: Restore: Read C1,C2: End If
Fade 25,.C1	C2
Until Colour(1)=C1 and Colour(9)=C2
Loop

'MARCH SETUP

Default
Edit
Data $46F, $FF
Data $PO.5F
Data $0.5FF
Data $POF, $F0
Data $FPE, $EE
Data -1,-1

Procedure BONE[X,Y,SX,SY,C]
Bar X-SX,Y-SY To X+SX,Y+SY
For R=1 To C
Circle X-SX,Y-SY,R
X-X+SY-R
X+SY-R
X+SY-R
Next

End Proc

Wicked isn't it? Watch out for the next Bone Demo (*): "Bone Demo III against Godzilla (*)" coming soon to an Amiga near you!

All for now, I hear Francois coming back, I have to stop writing this and do as if I was programming the compiler.

See you in the next Newsletter!

I wish you a merry christmas and a really happy new year full of dog food, bones, AMOS, compilers etc!

Bye Bye from France, Daïzy!

Note to all Australian male dogs (over 40cm high of course): did you ever think about some holidays in France? I would be delighted to welcome you here and take you up to my room. We could have some very nice moments together... Hummmmmmmm...

(*) Bone Demo II is a registered trade mark of Daisy Software Unlimited.

And Thus concludes Another edition in a story that's gone to the dogs!
AMOS File Selectors
A Tutorial & Comparison

With the vast volumes of mail I get each week, one of the most common sources of confusion that seems to appear is about the file requestor that comes with AMOS 1.2 and upwards. So to try and clarify things a bit, I have written this tutorial which will try and explain the functions of the gadgets on each of the 1.1 and 1.2? requesters. We will also make a quick comparison between the two.

Firstly we will give a brief run through on the V1.1 Requestor, this will be brief as it really is out of date! But a lot of it is also applies to the V1.2? Requestor.

AMOS V1.1 File Requestor

The AMOS 1.1 File Requestor.

The V1.1 file requestor had 6 main gadgets, These were the OK, QUIT, PARENT, SETDIR, UP ARROW and DOWN ARROW. (See Diagram Below)

Also depending on how many drives were connected, there were more gadgets added for each drive. This enabled you to select a different disk/drive quite easily.

At the top of the file requestor you had space for 3 lines of descriptive text, you would use this to ask a question or tell the user what the requester has appeared for, eg Please Select An IFF Picture.

Below this is the area where the file names and directories are listed. To select a file, you simply click on the file name twice with the left mouse button.

When you click once, the file selected's name will appear at the very bottom of the file selector, click again and the file selector will close and your file will be loaded. (Depending on how you called the file selector from your program)

Also where the file name appears, is the directory path and filter. When you click on directories then extended path will be displayed followed by the filter. The filter is usually "*", this means it will only show directories and file names that have a "*" in the name, eg program.amos, or picture.iff. You can of course specify how to filter the files, eg *.IFF - will only show directories and files that end in .IFF – AMOS will only show directories and files that end in .AMOS and if you leave it blank it will show every file and directory.

Now we will cover the main gadgets:

OK: - This is the same as the second set of colours can be a problem. The following simple modification will appear.

QUIT: - This simply quite the file requestor without selecting any files.

PARENT: - This gadget allows you to return one directory up at a time from within sub directories, so if you had a path of AMOS:samples:system/another_directory/ and you hit the parent button you would have a path of AMOS:samples:system/ and you would continue to climb out of sub directories until the ROOT directory is reached.

SETDIR: - This allows you to set the directory path so that next time you call up the file requestor you would automatically be in the directory that you selected with the SETDIR command.

UP ARROW: - This simply scrolls the directories and file names UPWARDS.

DOWN ARROW: - Like the Up Arrow, this scrolls the directories and file names DOWNWARDS.

With the V1.1 File Requestor, the Directories and File Names are automatically sorted as they are read in.

To see where the gadgets described above are on the requester refer to the V1.1 Diagram.

AMOS V1.2? File Requestor

The AMOS V1.2? File Requestor.

There were some major changes made to the old V1.1 File Requestor due to the suggestions and complaints from AMOS users. It looks somewhat similar, but thats where the similarity stops.

Take a minute to compare the two diagrams and how things have changed.

As you can see there are quite a few differences, there appears to be some gadgets missing and some new ones added. Confusing? No not really, its much better than the old V1.1!

Let's go over the changes and then cover the gadgets themselves. First of all the TEXT area at the top of the requester is the same as the PATH area.

Above this is the area where the file names and directories are listed. To select a file, you simply click on the file name twice with the left mouse button.

When you click once, the file selected's name will appear at the very bottom of the file selector, click again and the file selector will close and your file will be loaded. (Depending on how you called the file selector from your program)

Above where the file name appears, is the directory path and filter. When you click on directories then extended path will be displayed followed by the filter. The filter is usually "*", this means it will only show directories and file names that have a "*" in the name, eg program.amos, or picture.iff. You can of course specify how to filter the files, eg *.IFF - will only show directories and files that end in .IFF – AMOS will only show directories and files that end in .AMOS and if you leave it blank it will show every file and directory.

Now we will cover the main gadgets:

OK: - This is the same as the second set of colours can be a problem. The following simple modification will appear.

QUIT: - This simply quite the file requestor without selecting any files.

PARENT: - This gadget allows you to return one directory up at a time from within sub directories, so if you had a path of AMOS:samples:system/another_directory/ and you hit the parent button you would have a path of AMOS:samples:system/ and you would continue to climb out of sub directories until the ROOT directory is reached.

SORT: - This gadget allows you to sort the file names and directories according to the drive that the file is on, eg A:/ and B:/.

SETDIR: - This allows you to set the directory path so that next time you call up the file requestor you would automatically be in the directory that you selected with the SETDIR command.

UP ARROW: - This simply scrolls the directories and file names UPWARDS.

DOWN ARROW: - Like the Up Arrow, this scrolls the directories and file names DOWNWARDS.

With the V1.2? File Requestor, the Directories and File Names are automatically sorted as they are read in.

To see where the gadgets described above are on the requester refer to the V1.1 Diagram.

Adding To The Sprite Editor

If you have been having problems merging two sprite banks together, then these couple of simple changes to the sprite editor should make it much simpler!

First set the text buffer to 80000 bytes and then load the sprite editor into AMOS. Now find the line near the beginning of the program (should be line 1) which reads Set Buffer 11. Change it to read: Set Buffer 12. Now use the FIND TOP option and tell it to search for 1 As="L". This should be line 315 if you haven't changed anything. Change this line to read: If (A$='0 Or (A$='M').

Then cursor down about 19 line to the line that reads: Erase 1. Change it to read: If (A$='L').

Now if you press M you will be able to MERGE a second or third Set of Sprites/Bobs onto the end of the set already loaded into the Sprite editor.

And of course, by pressing L all old Sprites/Bobs will be erased and the new set loaded in their place!

YET ANOTHER MODIFICATION

Well that's one modification to the sprite editor, now let's do another! If you are working with sprites you would probably notice that sprites use a second set of 16 colours, but are drawn with the first set of 16 colours. This can of cause cause problems when designing sprites, because getting at the second set of colours can be a problem. The following simple modification will simply copy the first set of 16 colours into the second set of 16 colours. This happens when you press the "R" key.

Find the line (315) that we modified in the above example and insert this line before it: RGGColor: Screen:Screen 1:Wait Vbl

Now find the label that reads LOADSPRITES; and type the following lines before it:

RGBColor:
Screen:Screen 1:Wait Vbl
A=0:Repeat
Colour=A+16,Colour(A)
ldo A=A+16
Screen S
Return

And now you have yet another function in the sprite editor. Make sure you save your new version of the sprite editor to a backup disk. If you come up with some other new functions for inclusion to the sprite editor then drop me a line and we will publish them in up and coming issues of the Newsletter!
It's PD listing time! Over the last couple of months we have received quite a lot of new AUSSIE PD submissions! Which is of course great! It's good to see that you are managing to finish off your projects, and to quite a high standard I might add. Some of the games that we have received have been extremely good. As you can see from the listings we have about another 20 disks, all packed with games, utilities and demos.

I've got my favorites, but to be fair I really can't tell you which ones because we all have different tastes. We also have our regulars who contribute quite regularly, like a certain Husband & Wife Team who have only had AMOS for a little while but have so far written two full games! And both of excellent quality! This is going to be a short column this newsletter because the listings of PD are getting very BIG!

Just a quick word about how the disks are copied now. We have been getting quite a few people asking or write and say that their PD disks have got errors all over them. Well in 99% of cases that's ok! It's ok because we now use a BAM COPIER. What a BAM COPIER does is it first looks at the original disk and we will of course do you a new copy if the data does have errors. But errors still do occur, according to the BAM - Block Availability Map on the original disk. This explains why in 99% of cases, the errors are ok because they have never had data written to them. If you want to get rid of the errors and write more data to the disks, then you simply have to format a new disk and copy the files to it. When we are copying disks they are always verified, any disks that show errors while copying are automatically trashed and another copy is made. But errors still do occur, and we will of course do you a new copy if the data does have errors.

AUSSIEDISK Listings

**AA1**...Dark Angels MAP/SCREEN Editor
**AA2**...124 Instruments for GMC/Soundtracker
**AA3**...122 Instruments for GMC/Soundtracker
**AA4**...82 Instruments for GMC/Soundtracker
**AA5**...54 Instruments for GMC/Soundtracker
**AA6**...99 Instruments for GMC/Soundtracker
**AA7**...69 Instruments for GMC/Soundtracker
**AA8**...7 Soundtracker 2.3 Modules
**AA9**...3 Soundtracker 2.3 Modules
**AA10**...6 Soundtracker 2.3 Modules
**AA11**...10 Soundtracker 2.3 Modules
**AA12**...12 Soundtracker 2.3 Modules
**AA13**...10 Soundtracker 2.3 Modules
**AA14**...8 Soundtracker 2.3 Modules
**AA15**...9 Soundtracker 2.3 Modules
**AA16**...11 Soundtracker 2.3 Modules
**AA17**...6 Soundtracker 2.3 Modules
**AA18**...6 Soundtracker 2.3 Modules+ST2.3/ST2.5
**AA19**...72 Various Sound Effects
**AA20**...58 Various Sound Effects/Instruments & Voices
**AA21**...Soundtracker 2.4, Noisetracker, 6 modules
**AA22**...The Sausage Demo + 4 Modules (By Sausagel)
**AA23**...Musical Squares. Excellent Sliding square puzzle game
**AA24**...FLAME. A Sausage Shoot-Em-Up, fast action!
**AA25**...Puzzle Game. AMONOT-Light Cycles. Kamlzradi Kombat-Shoot-Em-Up, 9 Sound Samples.
**AA26**...TEXDEMO! Demo showing use of Autostart and Large Animations (A Must for Max Headroom Fans!)!
**AA27**...MAZERUNNER. (Excellent Maze Game) SUB-HUNT. LIGHTCYCLES. SHIPWALKER (Similar to Mazerunner, but not a game-Actual Maze Routinew Great starting point for a game like Dungeon Master etc)
**AA28**...AMOSI.21 + all new accessories. New Soundtracker/Noisetracker Converter/Player + More!
**AA29**...FONT UTILTY-Viewer/Mover, Very Handy!
**AA30**...Unarmd But Dangerous. Excellent Martial Arts Game with 33 screens!
**AA31**...KENO. Excellent Version Of The Casino Game.
**AA34**...Sample Workshop. Play Raw Sound Effects, Change Speed Etc & Then Convert Them To An AMOS Bank, A Must Have For Game Writers!
**AA35**...V8 Music Faced, A Musical CD Player/DEMO.
**AA36**...AMOS DEMO II. This Is The Totally NEW AMOS Demo! Some Excellent Special Effects!
**AA37**...Game Disk. Contains A Version Of Lightcycles Called TRAP, A Shootemup Program.
**AA38**...BATMAN DEMOS. (No Not THE Batman) Some Good DEMOS!
**AA39**...Asst/Games/Demos/Programs including...
Amastormer, Landline Rainbow Utility + More!
**AA40**...1942-The Game. Excellent version of the Popular Arcade Game Complete With Sample Sound From The Original Arcade Game. Needs 1 Meg.
**AA41**...The RACE. Furious Joystick Wiggling Action!
**AA42**...Modified AMOS Accessories including Sample bank maker, Soundtracker Converter, PAL Sprite Grabber, AMOS

British PD Disk Listings

**BA1**....GMC (Games Music Creator)
**BA2**....Fonts Disk 1-Contains 14 Fonts
**BA3**....Fonts Disk 2-Contains 13 Fonts
**BA4**....Fonts Disk 3-Contains 14 Fonts
**BA5**...Disk To Disk. Convert STOS (ST) Programs, Sprites etc
**BA6**...VIRUS X 4.0 Anti Virus Program
**BA7**...49 Sounds/Instruments + 32 Music ABKs
**BA8**...Treasure Search-Good Educational Game
**BA9**...New AMOS Demo V4
**BA10**...50 Assorted Sound Samples
**BA11**...32 Assorted Sound Samples
**BA12**...30 Assorted Sound Samples
**BA13**...24 Assorted Sound Samples
**BA14**...13 Assorted IPP Pictures
**BA15**...27 Assorted IPP Pictures
**BA16**...17 Assorted IPP Pictures
**BA17**...92 GMC Instruments
**BA18**...77 GMC Instruments
**BA19**...Microsoft Sound Sensitive Balls Demo
**BA20**...ARC Angel Demo
**BA21**...Word Square Solver Game
**BA22**..Fun School 3 DEMO-Not Yet AVAIL!
**BA23**...7 SoundTracker 2.4 Modules + Player
**BA24**...8 SoundTracker 2.4 Modules + Player
**BA25**...8 SoundTracker 2.4 Modules + Player
**BA26**...7 SoundTracker 2.4 Modules + Player
**BA27**...7 SoundTracker 2.4 Modules + Player
**BA28**...11 SoundTracker 2.4 Modules + Player
**BA29**...6 SoundTracker 2.4 Modules + Player
**BA30**...7 SoundTracker 2.4 Modules + Player
**BA31**...Crosby Screen Designer
**BA32**...Asst Procedures/Programs/Games
**BA33**...Pink Goes Ape-ARC Angel Demo2
**BA34**...Tile Game + 11 Songs
**BA35**...ARCHIVIST. Small General Purpose Database
**BA36**...Not Yet Available
**BA37**...ARC Angel Demo 3
**BA38**...Fonts Disk 4, Contains IPP Pictures which can be used as Sprite Alphabets in DEMOS & GAMES!
**BA39**...MUSIC#5 Contains 7 Songs.
**BA40**...MUSIC#6 Contains 7 Songs.
**BA41**...MUSIC#4 Contains 9 Songs.
**BA42**...MUSIC#5 Contains 11 Songs.
**BA43**...MUSIC#6 Contains 4 Songs.
**BA44**...MUSIC#7 Contains 7 Songs.
**BA45**...MUSIC#8 Contains 7 Songs.
**BA46**...MUSIC#9 Contains 7 Songs.
**BA47**...MUSIC#10 Contains 9 Songs.
**BA48**...MUSIC#11 Contains 8 Songs.
**BA49**...MUSIC#12 Contains 7 Songs.
**BA50**...MUSIC#13 Contains 8 Songs.
**BA51**...Weird Science DEMO. Includes high quality digitized photos out of Back To The Future II.
**BA52**...The Return of The Stinger. Forms Really Unlimited. If you have the need to design forms than this is for YOU! Includes Examples.
**BA53**...CROSS & STAVROS MEGA DEMO II.
**BA54**...MUSIC#12 Contains 7 Songs.
**BA55**...MUSIC#14 Contains 6 Songs.
**BA56**...MUSIC#15 Contains 6 Songs.
**BA57**...MUSIC#16 Contains 6 Songs.
**BA58**...MUSIC#17 Contains 6 Songs.
**BA59**...MUSIC#18 Contains 6 Songs.
**BA60**...MUSIC#19 Contains 6 Songs.

Continued On Page 9
AUSSIEDISK and British PD Library Listings

Ordering/Submitting PD Disks

When ordering PD Disks or AussieDisks, please make sure that you use the order form below. Please make sure that you fill in all the details, paying particular attention to filling in your REGO NUMBER. (It makes it so much quicker and easier when fulfilling your orders)

When you send in your orders for PD disks please mark your envelope "PD ORDER". Again it just makes it so much easier and quicker to sort out the volumes of mail we receive every day.

Also when submitting programs for inclusion in the AussieDisk library, please mark the padded bag "AUSSIEDISK SUBMISSION". Don't forget to put a quick note in explaining what is on the disk and what PD disk you want in return.

Quick word about submissions: Please don't send in your contributions procedure locked! The British PD Library is now rejecting any submissions that are locked and for very good reason! The whole idea of the library is to learn from what other people have done, you cannot of course do this if the program is locked! If you get some programs from the library and you see a routine that looks interesting, you would immediately want to find out how it was done! If the program is locked you can't! Anyway you should feel honoured if someone liked your routines that much, that they include them in their program. (It would of course be nice to be given a mention) But just keep all this in mind when submitting PD.

Keep those Programs, Sound Modules, Sound Effects and other bits and pieces rolling in! The Brits are way ahead with their library, they actually have heaps more because this listing only covers up to the beginning of December!

LAST MINUTE QUICKIE!
AUSTRALIAN AMOS STATISTICS

NSW --- 452 A500's -- 786
VIC --- 215 A1000-77
SA -- 97 A2000 -- 136
WA -- 72 A2500 -- 14
TAS -- 31 A3000 ---- 2
NT-4

AUSSIEDisk and British PD Library Listings

AMOS AUSSIEDisk & British PD Disk Order Form

NAME: ........................................
ADDRESS: ...................................
CITY: ........................................
STATE: ......................................
POSTCODE: ..................................
AMOS REGO NUMBER: .....................
DATE ORDER PLACED: . . . /

<table>
<thead>
<tr>
<th>CODE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CODE</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Send Order To: AMOS PD ORDER P.O. Box 253 Rydalmere, NSW 2116

AUSSIEDisk & British PD Pricing

1 DISK - $7.00
2 DISKS - 13.00
3 DISKS - $18.00
4 or More DISKS $5.00 Each

I Enclose $ . being in the form of Money Order( ) Personal/Bank Cheque( )