WhatIFF? Issue 1.06

This PDF has been generated by a proof of concept AG2PDF function. It's very basic so far and not even close to finished but here you go!

WhatIFF? For The Creative Amiga User

```
      <span class="shine">WhatIFF?<span class="fill">-Issue 1.06 August 2022</span></span>
                  For The Creative Amiga User

T ================================================= |T
       | ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|[L
       | _________________________________________________________[|
       |I __==___________  ___________     .  ,. _ .   __  T|
       ||[_j  L_I_I_I_I_j  L_I_I_I_I_j    /|/V||(g/|   ==  l|
       lI ________________________________  _____  _________I]
        |[__I_I_I_I_I_I_I_I_I_I_I_I_I_I_I_] [__I__] [_I_I_I_]|
        |[___I_I_I_I_I_I_I_I_I_I_I_I_I_L  I   ___   [_I_I_I_]|
        |[__I_I_I_I_I_I_I_I_I_I_I_I_I_I_L_I __I_]_  [_I_I_T ||
        |[___I_I_I_I_I_I_I_I_I_I_I_I_I____] [_I_I_] [___I_I_j|
        | [__I__I_________________I__L_]                    |
        |                                                    |  -Row
        l____________________________________________________j

                        ENTER MAGAZINE
```

Welcome to WhatIFF?

```
==============================================================================
                  -# WhatIFF? - Issue 1.06 - August 2022 #-
------------------------------------------------------------------------------
      --EDITORS DESK--   -MEET THE WRITERS--
      A few word from the current editor about the magazine.
      ---SHELL TALK---
      A place for WhatIFF? writers and readers to air out their Amiga
      laundry or just talk about anything Amiga related!
      -----REVIEWS----
      Reviews of hardware and software, old and new covered.
      -----GUIDES-----   -----ARTICLES------
      Guides and support to all things Amiga.
      -TALKING AMIGA--
      Hear from the people making hardware and software for the Amiga.
      -----LINKS------   ---AMIGA VENDORS---
      Amiga related links and Amiga vendor websites.
      -INVENTS CORNER-
      Kevin Saunders regular column for WhatIFF?
--------==================================================================-----
      NEXT ISSUE *All content copyright of the respective authors 2022*
--------==================================================================-----
```

From The Editor's Desk

==========
From The Editor's Desk - by: Timo Paul
==========

Welcome to issue 6 of WhatIff? and what an issue we have for you this time round. To start off with we welcome Ivan Sorensen to the WhatIFF? writing staff. For his first issue he has written two articles: Floppy Disks and You and Authenticity, the Amiga and You bringing his expertise and knowledge to the the magazine. Also just for this issue we also have two submissions from Carl Svensson's: ARexx Stems as Mock Data and Little Things That Made The Amiga Great. Even though his submissions are just for this issue we look forward to him writing for the magazine again the near future.

As if that is not enough we have reviews of the A500 Mini, Infinity Music Player v3, AmiSSL Classic 5.2 and The LightWave Book. The Brilliance and LightWave 101 tutorials continue with the latter look at the LW-Config file and how it can help in streamlining your work process.

John Scolieri is back with an article for Amiga OS4.1 users with a tutorial on reading and creating ADFs and Kevin Saunders updates us on his latest game projects regarding the Vampire series of computers. As you can see this issue is quite packed and this is all thanks the writers for their amazing contributions to the magazine.

Some of the more eagle eyed of you might have noticed a new section on the main page: Amiga Vendors this list will be updated each time a anew Amiga dealer is found. If you are selling Amiga goods and are not on the list then contact us so we can add your details.

I hope you will enjoy this issue and what it has to offer, it is incredible to think we have made it to issue 6 already. While there are other more professional printed Amiga magazines available I hope that we can offer something a bit different to them. If you have any suggestions or want to contribute to the magazine in anyway please contact us at:

submissions@whatiff.info

Free For All About The Amiga

==========
Shell Talk - by: Daniel Reimann
==========

How I Rediscovered My Floppy Disks

What are, among many other nice things, the advantages of having a baby? You get to know your surroundings from a completely new perspective! On our living room bookshelf, which is IKEA of course ;), there are two separate compartments on the bottom: one is filled with magazines, hiking maps and travel books, the other with my two Amiga floppy disk boxes and quite a few other boxes.

Currently, my child has developed two habits, which results in both shelves being completely cleared out :) He has a preference for hiking maps, which are unfolded with great pleasure and don't consist of only one piece. In addition, he also enjoys floppy disk boxes. There are two of them: the Verbatim "Deluxe" version, which offers a small plastic cover for each disk (like Boeder) and the Profex. Here is my complete collection:

Now I don't know if it's sacrilege to keep floppy disks not in professional plastic boxes/slipcases, but just in the original cardboard boxes. However, it was the trigger for me to dig back into old memories.

Vividly I can remember the various rummaging through boxes and walls in the local electronics store, in which the disks of various suppliers were presented. And every time the question came up: do I take the cheap no-name parts or do I treat myself to a box of the brand suppliers? As it turned out, I gladly reached for Boeder & Co and treated myself to a variety of colourful disks. Even in the 90s, one was already aware of sustainability, see KAO. And the no-name boxes were semi-professionally glued together and thus used as a kind of big box.

Now some of you may not believe it, but a considerable part of my floppy disk (collection) contains software from Aminet. Of course, like many others, I used a modem on my Amiga 1200! First, a 28.8k modem, which drove me crazy, because I never managed to get a noise-free dial-up, and later finally a 56k modem. At that time, in the mid-90s, I was studying and Aminet was the best place where I could get software quickly and for free - yes, at that time dial-up at home still cost something though ;)

Where is all the download software from Aminet? In the computer cabinet of course!. It felt like there was a new version of Voyager and Co every week and I could hardly keep up...

Are floppy disks now just an outdated greeting from times long past? I have to admit, it's been a while since I put the last floppy disk into the A1200 (but that also has to do with the current "computer offspring"). On the one hand, of course, they consume a lot of space. For the little bit that's on there...On the other hand, it's very pleasant to have an antithesis to this "WLAN/streaming/'file-shuffling' world. And even if it might seem contradictory regarding the 'wild copy times' on the Amiga: doesn't such a disk better symbolise the efforts and value behind the development of the software? In any case, my curiosity is aroused and I will be rummaging around in the boxes some more ...

Welcome to the reviews section!

==========

- Reviews -

The Lightwave 3D Book Review

It may be an old book but is this the book Amiga Lightwave 3D users have been looking for?

Infinity Music Player Review

Forget about Apple Music or Spotify, Infinity Music Player is on the Amiga and showing how a 30 computer can still compete with the next generation of computer for alk your MOD streaming needs.

AmiSSL 5.1 Classic Review

We take a quick look at the latest version of AmiSSL 5.1 for classic Amigas and see if it is worth your time updating to this version.

The A500 Mini Review

While WhatIFF? does not cover games, it would not be acceptable for us to not have a look at the closest we get to new Amiga computers in 2022.

The LightWave 3D Book Review

==========
The Lightwave 3D Book Review - by Timo Paul
==========

Product Name
The LightWave 3D Book - Tips, Techniques, and Ready-to-Use Objects
2nd Edition

Available From
Amazon/eBay/Used Book Stores

Price
$1-20

The Lightwave 3D Book Review
LightWave 3D is an very user friendly programme, especially when put against the likes of Imagine. However to truly get at the true power that LightWave affords a little guidance is needed. There are a large selection of books for the modern versions of LightWave centred on Windows and Mac OS computers, however there have not been any new books written about LightWave 3D and the Amiga.

The majority of the books that have recently been written about Amiga concentrate on games and the history of Commodore. Where it comes to guides or tutorials the options are a bit more limited. In the mid 90s after Lightwave 3D was released as a standalone programme there were a number of tutorial books written. I was lucky enough to find one that advertised in Amiga World and had it shipped over the States.

This book had some great tutorials and hints for beginners and veteran users alike. Since starting the LightWave 101 series I have been wanting to dive back into the book again to help me with the series but unfortunately the original book has been lost to time. Having had an unsuccessful search for it on the internet I decided to find an alternative book.

After much searching I realised that there are not many books that deal with the Amiga and Lightwave 3.5. However, after much an exhausting search I came across a book titled 'The Lightwave 3D Book' released in 1997.

This book peaked my interest due to two factors, it was created by the editors of LightWavePro Magazine which I had read a few times in the past, and they had often mentioned the Amiga and the book mentions LightWave 3.5 which as far as I know was only on the Amiga.

With my hopes high I ordered the book from Amazon for the princely sum of $10.00 plus shipping. It should be noted that this book can also be picked up for quite a bit cheaper as noted in the cover the of the book when it was sold for (Ahem) $1.00.

The Book
The book itself is roughly A4 size but a slightly wider, it comes in at 287 pages and includes a CD-Rom with textures, objects, plugins and applications (including ones for the Amiga!). It is a hefty book with thick pages and is printed in black & white.

Since this is a collection of guides from a magazine the contents are organised by subject instead of publishing date, as was done in the first edition. This makes it easier to jump in and out of the book and find something you need based on a particular topic.

The tutorials themselves are quite detailed and relatively easy to follow, in some instances there are grey scale photos but in general they are not needed due to the precise writing.

If you are a beginner to LightWave 3D this book however will not be for you. The articles presume a certain affinity and experience with LightWave 3D which is understandable since they come from a magazine that was solely dedicated to LightWave.

This brings me onto the writing style, while the topics covered are serious in nature, the writing tone is light with humour often sprinkled into the articles. This helps make something that could be quite dry reading into an enjoyable experience in itself.

The authors themselves are quite the heavyweights in the 3D animation world having worked on Babylon 5, seaQuest and RoboCop The Series to name just a few shows.

It makes for a fascinating read almost 30 years later as the animators detail the hurdles and tricks they used on some of the biggest shows in the 90s. Reading about them using our lovely Amiga in the process just makes it so much more enjoyable!

Even if you are not interested in learning about animating in LightWave or are a beginner the articles about the creation process are a great read in themselves. It is something that I would like to see more of featured in the Amiga history books. The Amiga realy was revolutionary with the Video Toaster and LightWave and deserves more recognition for what it contributed to TV and movies in the 90s.

For those of you who are interested in learning more about LightWave techniques then this book has a very good learning curve and each tutorial is explained in simple enough English as to not make it too hard to follow. With YouTube having become such a large part of life it is easy to forget how useful a guide book like The LightWave 3D book can be. You can move at your own pace and not have to worry about pausing or rewinding the video or even worse, the dreaded pop up adverts right when you are getting to the hard part of the video.

The CD
Yes,this book comes with CD and even has some Amiga programmes which still manages to surprise me.

The drawers on the CD are as follows:

DEMOS : Some videos/animations done the leading animators of the time.

DOCS : Contains all the Readme files for certain Scenes, Objects.

LW3_5 : Specifically for LightWave 3.5 and earlier, it contains all the Scenes/Objects/Images configured for the older versions of LightWave.

OTHER : Demo of LIGHT-ROM textures, images and objects.

PLUGINS : PowerView Plugins for PC and Amiga!

PROGRAMS : Useful programmes for the Amiga and PC.

SETUPS : Setup files for one of the tutorials lised in the book.

As you can see there is quite a bit of content for the Amiga on the CD-Rom which was a pleasant surprise. While the Amiga was very much alive for me in 1997 the rest of world had forgot about the Amiga and so it is nice to see the Amiga represented so well in the book.

Final Thoughts

If you are looking for a book to help you understand the basics of how to use LightWave, this is not for you. However if you already have a basic understanding of LightWave and have created some simple objects or animations then this book will probably suit you to a T. The topics covered are wide ranging and detailed enough so that you will not have too many questions. The book contains lot of Amiga centric articles as well which is always a plus. The book can be found quite easily on Amazon or other online bookstores, it can range from very cheap to reasonable.

There are not many Amiga tutorial books available anymore and if you want to learn more about LightWave then this is a book that I highly recommend.

Pros and Cons

+ Many topics
+ Easy to read writing style
+ Covers the Amiga
+ CD Rom included
- Not for first time users

Infinity Module Player Review

==========
Infinity Module Player Review - by: Timo Paul
==========

Product Name
Infinity Module Player (IMP) v3.355

Created By
Pawel Nowak

Available From
Aminet

Price
Free/Donation

Hardware/Software Requirements
68K OS 2.04+/PPC/MorphOS/i386-AROS
Network connected Amiga

Machine Used For Testing
Commodore Amiga 1200
Classic WB + 3.2 Kickstart
TF 1260

Infinity Module Player (IMP) Review
Listen (to) songs from (the) internet - over 168,000 files! (mods, SIDs, AHXs, XMs...), that is the first in the readme file that is on Aminet. The brain child of Polish programmer Pawel Nowak, IMP can be viewed as an Amiga version of streaming audio software like Spotify or Apple Music, but instead of an audio stream the music is downloaded straight to your Amiga and played, then when the song has finished another one will be downloaded and played and so on. Not streaming as we know it persay but I think something that is uniquely Amiga and brimming with potential.

Currently there are a number of internet radio stations playing Amiga music but it has been limited to the more powerful Amigas or PCs/Mac due to the need to decode the audio stream fast enough to be listenable.

Luckily a few pioneering individuals realised that it would be possible to stream mods directly to the Amiga meaning that the lower spec Amigas that were connected to the internet could enjoy 24/7 streaming of the latest and classic mods. Of the few programmes that have been released Infinity Module Player caught my eye or shall I say ear that is.

If you are a regular of EAB then you will probably already know about the IMP thread, which has the latest updates and fixes posted there first. If not, I recommend heading over to EAB and checking out the thread as it is interesting to see how the programme has evolved from something that was originally used mainly by Polish Amiga users to something that is being used by Amiga users all over the world.

This will be a short review and introduction to the Infinity Module Player (IMP).

Installation
IMP has gone through several changes since it's inception, starting as a programme that ran in the shell it now has a proper interface made up up several windows; a command menu, player control, scope, and a chat window.

What raises it above the other mod streamers is its ease of installation. Just download the software from Aminet, copy it to your location of choice and double click on the icon. By default there is tool icon so you will need to run it either from the Shell or click again on 'Ok' when the execute window pops up. You will need to have a TCP client running in the background but other than that, there is nothing else required. Upon first starting you are taken to a new custom screen which seems to be set at PAL Hires screen and asked to create a new user name, from then on the user name will be saved not requiring you to enter it again.

While it goes into a custom screen by default you are able to change it to a public screen and view it on Workbench. Other than these few steps that is all that is needed to run IMP.

In Use

Once you have the programme up and running you have four windows to contend with. The programme by default runs in a custom screen but this can be changed to run in any resolution or on the Workbench in a Public screen. I chose to use it on the Workbench since I run my Amiga in HIGHGFX: Super HiRes, which gives me plenty of desktop real estate to have all of IMP's windows and still have space to open other windows and move files around.

After you have created your user name the programme will not do anything until you click on the play button. This was the only issue that I had with the programme and it is probably just me. However I found it a little hard to locate the play button, I was looking for something more akin to the traditional Play/Pause button found on other music players. In IMP's case it is just a thin arrow with another button with two arrows next to it. Probably 99% of IMP users will not have any trouble with this but it was something that stuck out at me.

Getting back to the layout, I mentioned above the command menu, playback, scope and chat windows. One window in particular that I enjoyed was the 'SCOPE' window which shows the four channels and tracker patterns. Having the ability to see how the song is structured and plays, creates more personal connection with the music. When listening to MP3s or streaming music on PC/Macs you can't see how the structure of the music however thanks to the MOD format we can see how the music was made and as one who also makes music on the Amiga I find this most enjoyable. However for those who do not have the space for the window or are not interested it can be closed and re-opened again by clicking on 'SCOPE'.

Let's take a deeper look at the Menu Window, it is possible to control IMP via the menu bar as such all control is done via buttons. I will list each button and what it does:

```
CHAT   - Opens the chat window.
DISK   - Gives you access to other files held on the IMP server.
FAVS   - A list of the tracks that you favourited.
TOPS   - The most played files, click module name to play.
STASH  - A place to store your own MODS, only available to donators.
DISCO  - a funky screen with moving colours, simple but fun effect.
ZGR3D  - A small game that reminds me of Digger.
SCOPE  - Displays the module patterns.
HELP   - Did not seem to do anything.
SEARCH - Find a particular module.
SAVE   - Save the current MOD to your Amiga.
TYPE   - Set which format of songs you want to listen to: Modules, Sids,
         AHX, XM, and DBM.
MODE   - Choose if you want to play only the songs you liked, only with
       likes, ones you searched, never played or all songs.
PREFS  - Palette, Screen, Fonts, Switxh Workbench, LED Filter: On/Off
```

TEST - Tests your network connection speed.

As you can see there is a lot functionality to IMP that goes beyond a simple MOD streamer. It pays to deep dive into the menu and learn how it works and what is possible.

Final Thoughts

The idea of a MOD streamer is very appealing, while I wish I had the time to search all the MODs myself the reality is that there is just too many and not enough time. Infinity Module Player is the Spotify for Amiga MODs, it makes listening to a large library of music as simple as clicking a button. The version that I reviewed v3.355 was quite stable and I only experienced one issue where the whole screen became garbled in red and yellow. However after a quick reboot and everything was back to normal, the only effect I had was that I lost part of this review.

This software has a mainstay on my Amiga, in fact somuch so that I have ended up buying a mini hifi and have it hooked up to the Amiga 24/7, streaming music most of the day, much tothe annoyance of my wife.

The ease of use, the multiple functionality, the social aspect and the massive library of MODs makes this one of the best pieces of software I have reviewed of recent. It is fantastic that the programme is available for free, though I recommend donating to the author for his hard work.

Bottom line, download this now, if you don't have a network connected Amiga, get it connected, and hook your Amiga up to proper speakers and boogy on down to the tunes. Right time for me to bust some moves!

Pros and Cons

+ Easy to install
+ Loads of functionality
+ Stable
+ Free

AmiSSL 5 Review

==========
AmiSSL v5 Review - by: Timo Paul
==========

Product Name
AmiSSL v5.2

Created By
AmiSSL Open Source Team

Available From
Aminet

Price
Free

Hardware/Software Requirements
68K Amiga OS 3.0+/68020+
Network connected Amiga

Machine Used For Testing
Commodore Amiga 1200
Classic WB + 3.2 Kickstart
TF 1260

What Is AmiSSL
If you are new to AmiSSL, in a nutshell it is a collaborative effort to develop OpenSSL as a shared library for all Amiga systems (Amiga OS, Morph OS, AROS, and so forth). SSL stands for Secure Sockets Layer and is the standard technology used for keeping an internet connection secure and protecting any sensitive data being sent between two systems. As such it is now a requirement for any and all websites so if you plan to surf the web on your classic Amiga then AmiSSL is a necessity. Lucky for us the library is being distributed for free via Aminet or the AmiSSL website.

In Use
Installation is taken care of with the standard Workbench installer and it is simple and quick. One thing that is different from the last version of AmiSSL v4 is that users with the TF1260 060LC cards can now install the 060 version. Previously this would cause Amigas with the LC060 chip to freeze, the reason being the LC has no FPU and the v4 of AmiSSL was most likely looking for a full 060 with FPU.

I am happy to report that when I installed v5 and chose 060 the installtion went smoothly. More so when I tested the latest version of AmiSSL with AmiGemini which requires AmiSSL there was no crash and it was visibly faster than previously.

It is great to see AmiSSL supporting more hardware and in particular the Terrible Fire cards. If you have an TF1260 with the LC 060 card and wantt to browse Gemini capsules or use the latest version of IBrowse then you will need this.

Final Thoughts
As a review this is very short and lacking on details however all you really need to know is that if you are planning to surf the Internet with the latest browsers or AmiGemini, then you need this. You don't have to choose (68020/030/040/080 or MorphOS) anymore if you have a TF1260LC card as it seems the 060s without FPUs are also supported.

A500 Mini Review

==========
A500 Mini Review - by: Timo Paul
==========

Product Name
The A500 Mini

Created By
Retro Games - www.retrogames.biz

Available From
Any electronics store

Price
$119.99

Hardware/Software Requirements
HDMI TV/Monitor

The First New Amiga Since 1996!
Well calling the The A500 Mini as a true new Amiga might be stretching the truth a bit but since we have not had a real Amiga since 1996 beggars can't be choosers as the old saying goes. You may be wondering why we are reviewing the A500 Mini in WhatIFF? since the magazine does not cover games, a good question. Well to be perfectly honest the opportunity to review new Amiga hardware in whatever form does not come up often, so it was decided to review the mini A500 in the WhatIFF? way.

First Impressions
The box reminds you right away of the cartoon classics pack that was released in the UK, the colour scheme and layout carry a heavy load of nostalgia. Upon opening the box you are presented with the smallest Amiga to date, a lovingly recreated A500. Under the mini computer you can find the mouse and control pad, as you have probably already seen in the photos of other reviews the mouse is a slightly smaller re-creation of the classic tank mouse that came with the A500s. The pad is reminiscient of the CD32 pad, this might divide opinion, personally I liked the CD32 pad as it was big and chunky and fit my hands nicely. The A500 Mini version however is a bit smaller and can feel a little cramped after long gaming sessions.

Taking a closer look at the A500 itself, the attention to detail is a delight to behold, all the original lines, font and keyboard details are there to take you back to your youth. Looking at the side you have a limited recreation of the floppy disk and on the right the lines of the casing joints, unfortunately the side slot cover is missing. A few others have mentioned this and I have to concur that it seems like a missed opportunity with the floppy disk area, it would have been nice to have an USB slot there when using WHDLoad games.

On the back you have the HDMI, USB-C, and three USB ports, right away you feel this is a little limited considering if you want to have an external keyboard, mouse, joypad and additional games. The power switch is on the back and is a simple push button. Upon plugging everything in and pushing it the red powerlight comes on bringing back again a surprising rush of memories, who ever thought seeing a rectangle red LED could be so emotive.

Starting Up
After pressing the power button you have a few seconds of a black screen followed by a white screen with the A500 logo and a quick

bounce of a half boing ball. Simple but slick and effective. The rest of your A500 mini experience takes place in the game carosele move to the left or right to highlight your game. Press the A button red start a game, you also have 4 game save slots that can be accessed by pushing down, for those extra hard games.

To access the Options press Menu on the control pad, from here you can change the display optons, mouse sensitivity, music volume, power LED settings, language, advanced options such as 50 or 60 hz and expert mode. When you use WHDLoad games you will be able to change more settings such JIT, changing the way the Copper works and more. The setting options are quite encompassing, going beyond the typical settings offered with other mini consoles and for this I commend the creators.

For this review we will not cover the games because what may be junk for one is gold for another and we do not review games here. However we will look at what you can do beyond the games and see if the A500 could be a nice alternative to buying a 'real' Amiga. As stated above it is possible to load WHDLoad games but what if you want to use the Amiga for more than games, is that possible with the A500 Mini?

Beyond Games

First you will need to prepare a USB thumbdrive, update the A500 Mini to the latest firmware and prepare an HDF file. At the moment the A500 Mini does not officially support HDF files but with the latest firmware update ADF support is now included. To access your HDF file you will need to bring the Amiberry configure screen up and this can be done by editing one of the .uae files. The easiest way to do this is to download any WHDLoad game and keep it in the .lha format. Play the game once on the A500 Mini and then shutdown the mini. Take the USB drive from your mini and look for the game you just played, for example if you downloaded and played Lemmings.lha, look for the file and below it you will see Lemmings_ol.uae, copy this file over to your computer and rename it to .txt. Now you can edit it as a text file, next you will need to add four lines to the top of the text file:

```
amiberry.open_gui=F12
                    amiberry.save_state_on_exit=true
amiberry.ignore_gui=false
use_gui=yes
```

Save the file and put the extension back to .uae, most of your probably already know what these four lines will do, open the Amiberry configure screen. Now when you run this game, instead of the game starting the Amiberry screen will pop up, from here you can change the settings like you would as if you were running in WinUAE or UAE for Mac. Change the CPU, chipset, add floppy disks or ... and this is what we are looking for add a HDF file. For my experiment I used ClassicWB LITE, and I am happy to say it worked. From here I was able to run Workbench and use the A500 Mini for much more than games. You will need to find a good keyboard and probably a USB hub but this proves that with a little work the A500 Mini could become a replacement Amiga.

What I described above was just a simple example and if you are not comfortable changing settings or have not used UAE before then you might want to do this with someone else who has experience with Amiga emulation software.

Final Thoughts

The A500 Mini is a well planned and excuted minature Amiga, it is easy to setup for beginners, comes with all the necessary utilities, connects to your modern TV and has the ability of add games with the minimum of effort and fuss. As it stands it is a great option for playing Amiga games on your TV or with family. It is particularly relevant today with the prices of Amiga hardware skyrocketing and existing hardware becoming more fragile and rare. If you want to

introduce your children to the joys of Amiga gaming without having to worry if you will need to take out a loan to pay for repairs if they become too excited then the A500 Mini is a great option.

Beyond the game playing the fact that it can be hacked to use HDF files tips it from being a good buy to a fantastic purchase. In essence you have a fully useable Amiga that has USB ports and can be plugged into a modern display without any extra hardware. Great for the beginner and veteran Amiga user alike.

Pros and Cons
+ Easy to setup
+ Ability to add games and HDF
+ Relatively cheap compared to real hardware
+ Looks great!
- Lack of USB ports
- No working internal keyboard

Welcome to the guides section!

==========

- Tutorials/Guides -

ADFs With Amiga OS4.1

John Scolieri takes us through the creation and reading of ADFs when using Amiga OS4.1.

LightWave 3.5 - 101 Configuration Files

In our fifth installment of the Lightwave 101 series we take a look at the LW-config and MOD-config files and see how editing them can make using Lightwave much easier.

Brilliance - 101 Stencil Tool

Already we have reached part 5 of the Brilliance 101 series and for this part we will be looking at how to use the Stencil Tool.

Floppy Disks and You

Ivan Sorensen takes us through the pros and cons of using floppy disks Gotek drives.

Chris Edwards - A1200 FPU Installation

Chris Edwards takes us through installing an FPU onto an Amiga 1200 motherboard and explains the pros and cons

ARexx Stems As Mock Data

Carl Svensson provides a tutorial on ARexx that can be saved out to the Amiga and used as a programme in ARexx.

ADFs With Amiga OS4.1

```
==============================================================================
        How to read/ write ADF's with AmigaOS4.1  - by: John Scolieri
==============================================================================
```

How to read/ write ADF's with AmigaOS4.1

Amiga Disk Files,or more commonly known as ADF's, are used by Amiga users to backup their original floppies as well as disk virtualization i.e. access an ADF files on Workbenches if it were a physical drive.

When I first bought my AmigaOne X1000 motherboard over a decade ago my trusty Amiga 1200 at the time had some hardware issues I couldn't figure out and a ton of disks which sadly were never backed up...I mean I knew I had my 1200 hard drive backed up but never thought about backing up my disks until my Amiga broke down. My X1000 didn't have a connector to directly use my external Amiga disk drive so in order to use my Amiga disks I had to purchase a Catweasel floppy controller from Individual Computers (icomp.de) which allowed me to use a commonly used PC floppy drive in my X1000 as if it were an Amiga 3.5 drive. With the 3.5" drive installed the next step was to install SuperDiskImage by Ian Gledhill over at his website (http://www.retroreview.com/iang/Catweasel/). Using SuperDiskImage is quite straight forward as follows:

To create an ADF file from a real 3.5" floppy-

1- Insert disk into 3.5" drive
2- Select which drive you want to use (Unit 0 or Unit1)
3- Click the 'Read Disk' button at top of screen
4- The blocks at mid screen will turn from black to green after being successfully read. Note: If a block cannot be successfully read it will appear red.
5- Once all blocks have been read remove the master disk from drive and save the new ADF file by typing the name of the disk into the 'Image' requester (don't forget to add the '.adf' at the end of the file name) and choose the drive/ partition/ folder to save to.

To copy an ADF file to a real 3.5" floppy-

1- Follow items 1 & 2 as above.
2- Use the file requester to select the file from your hard drive/removable media etc.
3- Click the 'Write Disk' button at top of screen
4- The image file selected will be copied track by track to the selected disk.

If you have some copy protected disks you want to backup you need to do the following:

1- Open a text editor to edit DEVS:NSDPatch.cfg
2- Add the following lines:

```
DEVICE trackdisk.device UNIT 0 MAPTODEVICE catweasel.device MAPTOUNIT 0 MACHINE=machinetype
DEVICE trackdisk.device UNIT 1 MAPTODEVICE catweasel.device MAPTOUNIT 1 MACHINE=machinetype
```

3- Install NSDPatch from aminet (if you haven't done so already)
4- Copy your SYS:Storage/DOSDrivers/TD0 to SYS:Storage/DOSDrivers/DF0
5- Reboot

If you are into using emulators on your AmigaOS4 system the Catweasel allows you to use your original Commodore 64/128, MS-DOS, ATARI, Texas Instruments, Apple Macintosh, Acorn BBC, Sinclair Spectrum, Amstrad CPC,

Tatung Einstein and Triumph-Adler Alphatronic disks! Of course you still need working drives from each system.

Many thanks to Individual Computers for creating the catweasel cards and to Ian Gledhill for creating SuperDiskImage.

Cheers,
G. Scolieri

LightWave 101 - Editing Configurations Files

```
==============================================================================
      LightWave 101 - Part 5 - by: Timo Paul
==============================================================================

  What Are LW-config & MOD-config?
  Welcome to the fifth installment in the LightWave 101 series, for this
  issue we will be looking at the two config files: LW-config (LightWave)
  and MOD-config (Modeler), both files can be found in the Toaster/3D
  directory.  The tutorial will be split into two parts, this part we will
  concentrate on the LW-config file and next time on the MOD-config file.

  If you remember back to the first installment in this series we looked
  at getting your assigns setup so LightWave would load correctly and
  allow you to access to the surfaces, objects, scenes etc. without
  causing issues.

  This is tutorial will go beyond setting up your assigns.  When you load
  LightWave and open a file requster you may have noticed that the choice
  of drives at the top may not be the same naming convention as your
  drives, for example LightWwave's the default device names are:

  Parent | DF0 | HD0 | HD1 | RAM

  In my instance the drives are not named HD0 and HD1, they are DH0 and
  DH1.  As such if I click on the HD0 an error requestor will pop up,
  well this can be fixed by editing the LW-config and MOD-config files.

  LightWave Config file
  Let's have a look at the LW-Config file first, below is a listing of my
  config file and I will show you what I need to fix.

  LWCO
  0
  -The two lines above are file identifiers and should not be changed.

  FileReqPreset1 DF0 DF0:
  FileReqPreset2 HD0 HD0:
  FileReqPreset3 HD1 HD1:
  FileReqPreset4 RAM RAM:
  -The FileReqPreset refers to the four buttons that can be seen at the
   top of all LightWave requesters.  In this instance they are pointing
   to the wrong drives, so I need to change them to DH0 DH0: and DH1 DH1:

  ScenesDirectory 3D:Scenes
  ObjectsDirectory 3D:Objects
  HierarchiesDirectory 3D:Scenes
  SurfacesDirectory 3D:Surfaces
  ImagesDirectory 3D:Images
  OutputDirectory 3D:Images
  FramestoresDirectory 3D:Images
  MotionsDirectory 3D:Motions
  EnvelopesDirectory 3D:Envelopes
  PreviewsDirectory 3D:Previews
  -The 10 lines above tell LightWave where to look when you click on
   Scene, Objects, Surfaces, Images, Motions, Envelopes and Previews.
   Currently they are pointing to 'fake logical volume' called 3D:, this
   was created in the 'user-startup' file.  The basic install of LightWave
   however will be written as such:

    ScenesDirectory DH1:Toaster/3D/Scenes
 ObjectsDirectory DH1:Toaster/3D/Objects
 HierarchiesDirectory DH1:Toaster/3D/Scenes
```

SurfacesDirectory DH1:Toaster/3D/Surfaces
ImagesDirectory DH1:Toaster/3D/Images
OutputDirectory DH1:Toaster/3D/Images
FramestoresDirectory DH1:Toaster/3D/Images
MotionsDirectory DH1:Toaster/3D/Motions
EnvelopesDirectory DH1:Toaster/3D/Envelopes
PreviewsDirectory DH1:Toaster/3D/Previews

StatusFilename (none)
-Back when we did not have powerful computers and render farms were common this was an option used for render farm software.

DefaultTension 0.000000
-Changing this can cause any generated keyframe to have tension value set to this number (-1 to 1), useful if you do a lot of logo work.

DefaultSegmentMemory 2200000
-(Camera panel) This number is used to how much segement memory LightWave will use when rendering images. It is set quite low due to Amigas not having much memory back in the day but with most Amigas having at least 64mb or more you can set this to a higher number.

DefaultZoomFactor 3.200000
-Default setting for camera lens when a new scene is started, the default creates a 24mm lens.

DefaultOverlay 0
-This interestingly is a feature that was requested by the seaQuest animators, the value will determine if LightWave's Data Overlay option will be on (1) or off (0) by default. If the setting is on, you will be asked if you want the scene name put into the Data Overlay label.

FrameEndBeep 0
-Similar to DataOverlay, 0 will be the default Frame End Beep (scene panel) off whenver you start LightWave and 1 will default it on.

RenderDisplayDevice 0
-Each time you exit LightWave the current settings will be written. Essentially it determines what the Render Display will be set to the next time you run LightWave. The options are: None (0), Toaster (1) 6-bit HAM (2), 8-bit HAM (3) and Picasso II (4).

RecordSetup1 (none)
RecordSetup2 (none)
RecordCommand (none)
RecordDelay 0.000000
FirstFrameDelay 0.000000
-Upon exiting LightWave the five lines above are written, they refer to text entered for Record Setup 1, Record Setup 2, Record Command, Frame Record Delay and Extra First Delay. These are useful when recording by single frame to video device.

DefaultLayoutGrid 4
-This is not written by LightWave but it sets the default settings for Layout Grid, the options being Off (0) to 16x16 (8).

AutoKeyAdjust 0
-Affecting the Auto Key Adjust button in the Options Panel, 0 is the button is off and 1 means the button is turned on. When you modify anything on a keyframe it will automatically create a new keyframe. Great for creating a still image but can wreck havok if you are doing an animation.

ExpertMode 0
-A redundant feature, however when it was turned off (0) you would be warned each time you selected a raytracing button, telling you that it would take a long time to render.

As you can see the LW-config file offers a lot of customisation, allowing you to setup LightWave how you want it. Next time we will look at the MOD-config file and see how we can customise the Modeler.

Brilliance 101 - Reflection and Gradient

==========
Brilliance 101 Part 5 - by: Timo Paul
==========

Stencils Galore!
Hello and welcome back to another installment of the Brilliance 101 tutorial series, this issue we will be looking at using the Stencil tool in Brilliance. First let's answer the question, 'Why would you want to use a stencil in Brilliance?', well Stencils provide an easy way to mask off designated areas of your workspace or screen to prevent them from being painted over. Sounds pretty neat right? Before we start please make sure you are in the HAM version of Brilliance, the creation of a stencil mask differs slightly from the Register mode so we will start with the HAM version of Brilliance for this tutorial.

Go to the Gradient menu and choose gradient 2, we choose this one as it has a large colour pallete and will help in highlightin Variance. Set Spread to 10 or higher, select Filled Rectangle tool and then choose Hightlight fromthe Draw Modes menu and drag out a large rectangle on your work screen. It is not important but you need to set the highlight position, anywhere is fine but for this tutorial we just put it in the middle of the box.

Next bring up the Stencil menu remembering to right mouse click on the Stencil tool. We won't need any other tools at the moment so you close the rest of the tools. Now that you are in the Stencil tool click on the Select button, you will notice the pointer will turn into an eyedropper. From here find the red area of the box that you drew out, click on the reddest part. Now that you have finished that part check the Variance settings are all set to 10. The manual explains Varience best so I will use it's explanation:

"Variance sets the acceptable range of colours, on either side of the selected colour, that will be incorporated into the mask."

The Variance settings refer to the Hue, Sauration and Value and we have set them at 10% which is pretty tight range, have a play around with the settings later on to see the different Varience settings will have on yor Stencil.

For now let's do a quick test, click on Draw Modes and double check that 'Color' is selected. Select black from the pallete and choosem the Filled Rectangle tool again, draw out a black box completely covering the gradient box. You will see right away that only the red part of the box shows up, congratulations you have done your first Stencil Mask in Brilliance.

Next Step
If you haven't done so yet, press U on the keyboard or click on UNDO to remove the black box covering your gradient box. Go back to the Stencil Tool box and set the 'Variance' to 20, click on 'Select' and choose the same red colour again. Click on 'Colors' in the 'Stencil Mask' tool, then draw the black box over your gradient box again, you will notice that there are more colours now showing. The Variance settings are much more useful when using the HAM version of Brilliance due to more colours being available.

So far we have used the Stencil in Replace mode, which means that we have been creating a new stencil each time. If you have a quick look at the Stencil tool you will see there are three options:

Replace

Add
Subtract

It is possible to use the Add mode to include area of different hues or the Replace mode to remove colours from the stencil mask. But to move on, press U or click on UNDO again and let's try one more trick with the Stencil tool for this part. Once you are back in the Stencil toolbar click on 'Invert', drag another black box and you will see this time only the red colour is removed with the other colours remaing.

One thing to note is that because we are working in HAM mode the stencil works with a much wider range of colours instead of a single colour, as such the active stencil colours are not shown. In the register version of Brilliance the stencil colours are shown with an oval.

Undo the black box again and delete the stencil mask by clicking on 'Delete' in the Stencil menu. Now clukc on 'Lasso', locate the area that contains the colours you want to isolate, draw a line around them. To see which colours became part ofthe mask, draw a black box over the gradient.

So far we have looked at how to create a stencil mask from the colours that are already on the screen. However Brilliance also allows you to easily draw the stencil mask onto the screen. Let's take a look at that next.

Drawing a Stencil
First you will need to delete your stencil, like before. After you have done that right mouse click on the 'Draw Mode' tool to bring it up and click on 'Stencil'. Choose a filled draw such as a 'Filled Rectangle' or 'Filled Shape', draw any shape, maybe your name if it is not too long. Remember if you want to remove a part of your stencil you can do this by holding down the right mouse button and drawing your desired shape. Once you have drawn your stencil, choose 'Color' in the 'Draw Mode' tool and once again select the colour black and draw a box over your entire gradient. This time you should see only your stencil appear filled with the gradient. Very easy isn't it?

Thank you for having joined me on this Brilliance tutorial, good luck with your artwork and if you have some finished pieces please send them into the Amiga Art Contest 2022, you can find more at:

www.amigaartwork.com

Until next time, keep creating and see you in the October issue for our next Brilliance installment.

ARexx Stems as Mock Data

```
==============================================================================
       Stems as Mock Data Structures in REXX - by: Carl Svensson
==============================================================================

/*****************************************************************************/

                Stems as mock data structures in REXX
             ==========================================
                  An experimental "Tutorial as Code"
               (c) 2021 Carl Svensson | www.datagubbe.se
            Tested in ARexx 1.15, Regina REXX 3.6, BRexx 2.1.9

 The REXX language was invented at IBM by Mike Cowlishaw and has since
 been ported from IBM's mainframes to many various platforms, including DOS
 and OS/2. In 1987 it was ported to the Amiga range of computers (as ARexx)
 by William S. Hawes. Today, REXX is available on a multitude of platforms
 thanks to the portable interpreters Regina and BRexx.

 REXX itself has no particular data structures as supplied in modern
 scripting languages such as JavaScript and Python. It provides variables
 and a single stack, but by using what could be described as "syntactic
 sugar" for manipulating the internal symbol table, it enables something
 called stems and compound variables.

 The aim of this tutorial is not to introduce the basics of the REXX
 language, but rather to explain a few concepts that are, AFAIK, unique to
 the REXX family of languages.

 Even though the tutorial has a stupendously pretentious name, the
 hope is that it will be instructive enough to explain how combining
 stems, subroutines and evaluation of strings as code can substitute the
 need for many other language constructs that are not immediately available
 in REXX.

 This tutorial comes with ABSOLUTELY NO WARRANTY and is licensed under
 the CC BY-NC-ND 4.0 license.
 For details, see https://creativecommons.org/licenses/by-nc-nd/4.0/

*****************************************************************************/


/* ========================================================================== */
/*                      Variables and the symbol table.                       */
/* ========================================================================== */

say foobar                      /* An uninitialized symbol will evaluate   */
                                /* to an uppercase string. This will print */
                                /* FOOBAR, as will henceforth be denoted   */
                                /* by the arrow "--->" in this tutorial:   */
                                /* ---> FOOBAR                             */

foobar = 42                     /* Initialize the variable foobar to       */
                                /* create an entry in the symbol table.    */

say foobar                      /* ---> 42                                 */

drop foobar                     /* Remove the variable from the            */
                                /* symbol table.                           */

say foobar                      /* ---> FOOBAR                             */
```

```
/* ========================================================================== */
/*                 Stems, compounds and the symbol table.                     */
/* ========================================================================== */

mystem. = 31337                 /* The stem mystem. is initialized         */
                                /* with the default value 31337.           */
                                /* Stems are not like lists or arrays, but */
                                /* rather syntactic sugar for working with */
                                /* the REXX symbol table - I.E., declaring */
                                /* variables.                              */
                                /* As such, they can be used to implement  */
                                /* or at least approximate the behavior of */
                                /* common structures in other languages.   */

say mystem.                     /* ---> 31337                              */
                                /* The value of mystem. itself is its      */
                                /* default value.                          */

say mystem.1                    /* ---> 31337                              */
                                /* The compound variable mystem.1          */
                                /* is now initialized to the stem's        */
                                /* default value. The stem's name is       */
                                /* mystem. and the 'index', 1, is called   */
                                /* a 'tail' in REXX.                       */

mystem.2 = 3                    /* Initialize compound variable mystem.2   */
                                /* with the value 3. All new compounds     */
                                /* get new individual entries in REXX's    */
                                /* symbol table.                           */

say mystem.2                    /* ---> 3                                  */

say mystem.1 + mystem.2         /* ---> 31340                              */

drop mystem.1                   /* Drop the compound mystem.1              */
                                /* from the symbol table.                  */

say mystem.1                    /* In ARexx and Regina REXX, this will     */
                                /* result in a completely dropped symbol,  */
                                /* which will evaluate to an uppercase     */
                                /* string:                                 */
                                /* ---> MYSTEM.1                           */
                                /* In BRexx it will result in reverting to */
                                /* the stem's default value:               */
                                /* ---> 31337                              */
                                /* This tutorial remains agnostic as to    */
                                /* which behavior is "correct".            */

mystem.foo = "Hello"            /* Compounds can be of any type            */
                                /* both in name and value, regardless      */
                                /* of the stem default. We can write       */
                                /* foo in lowercase for convenience,       */
                                /* but resolution will really be in        */
                                /* uppercase.                              */

say mystem.foo                  /* ---> Hello                              */

tail = "FOO"                    /* Initialize a normal string variable     */
                                /* intended for tail substitution.         */
                                /* Note the uppercase!                     */

say mystem.tail                 /* ---> Hello                              */
                                /* REXX resolves the value of the          */
                                /* tail before resolving the compound,     */
                                /* meaning we can dynamically access       */
                                /* arbitrary tails using a variable.       */
                                /* This is called tail substitution.       */
```

```
tail = foo                      /* Since uninitialized symbols evaluate    */
                                /* to uppercase strings, we can assign an  */
                                /* unquoted value to a variable we want to */
                                /* use for tail substitution.              */

say tail                        /* ---> FOO                                */
say mystem.tail                 /* ---> Hello                              */

mystem.bar = 12                 /* Here we create a clash:                 */
bar = "FOO"                     /* The tail BAR contains the value 12,     */
                                /* but we assign the value FOO to the      */
                                /* variable BAR.                           */

say mystem.bar                  /* ---> Hello                              */
                                /* REXX tail substitutes BAR with 'FOO',   */
                                /* giving us the value for MYSTEM.FOO.     */

drop bar                        /* Let's remove BAR from the symbol table. */
say mystem.bar                  /* ---> 12                                 */
                                /* Since we dropped BAR, MYSTEM.BAR now    */
                                /* resolves to the intended value.         */

newstem. = mystem.              /* It's not possible to create a copy of   */
                                /* or reference to a stem by assignment.   */
                                /* In this assignment, we simply create a  */
                                /* newstem. stem with the default value of */
                                /* 31337, since that's the actual value of */
                                /* mystem.                                 */
say newstem.foo                 /* ---> 31337                              */
newstem.foo = "Baz!"            /* This will only affect newstem.          */
say newstem.foo                 /* ---> Baz!                               */
say mystem.foo                  /* ---> Hello                              */
newstem.foo = mystem.foo        /* All values we wish to retain must be    */
                                /* copied explicitly. Remember, a compound */
                                /* will get a unique entry in the symbol   */
                                /* table!                                  */
say newstem.foo                 /* --> Hello                               */

x = 31; y = 13; z = 73          /* Tails can be chained to create multi-   */
coords.x.y.z = 23               /* dimensional stems.                      */
say coords.x.y.z                /* ---> 23                                 */
say coords.31.13.73             /* ---> 23                                 */

array.0 = 0                     /* A common pattern in REXX is to store    */
                                /* the number of stem elements in tail 0.  */

do i = 1 to 5                   /* When populating a stem intended for use */
  array.i = "I am "i"!"         /* as an array, we must keep track of the  */
  array.0 = i                   /* <=== element count                      */
end                             /* ourselves.                              */

say array.0                     /* ---> 5                                  */

do i = 1 to array.0
  say array.i                   /* ---> I am 1! ... I am 5!                */
end

/* ===================================================================== */
/*                              Interpret.                               */
/* ===================================================================== */

interpret "say 'Hello!'"       /* ---> Hello!                              */
                               /* INTERPRET allows for the interpretation  */
                               /* of an arbitrary string as REXX code,     */
                               /* much like eval() in JavaScript.          */
```

```
interpret "say array.1"       /* ---> I am 1!                            */
                              /* The global scope is accessible.         */

interpret "mystem.baz = 10"   /* The global scope can be manipulated.    */
say mystem.baz                /* ---> 10                                 */

anum = 1                      /* As evident from these lines and the ones */
interpret "bnum = 2"          /* just above, we can manipulate both      */
                              /* normal and compound variables.          */
interpret "say anum + bnum"   /* ---> 3                                  */
say bnum                      /* ---> 2                                  */

stem = "mystem"               /* We can also use INTERPRET to dynamically */
tail = "foo"                  /* access compounds based on the values of  */
                              /* variables:                              */
interpret "say "stem"."tail   /* ---> Hello                              */

/* ===================================================================== */
/*                          Labels and procedures.                       */
/* ===================================================================== */

my_label:                      /* A label. Without any special control   */
                               /* flow, this will simply be a no-op and  */
                               /* the code following it will execute     */
                               /* normally, just like a label in C.      */

say "Below my_label."          /* ---> Below my_label.                   */

call skip_here                 /* Here we call the label skip_here, as   */
                               /* declared below.                        */

say "Between call and label."  /* This line will not be executed.        */

skip_here:                     /* The label called above.                */
say "We are now here!"         /* ---> We are now here!                  */

call subroutine                /* Call the label "subroutine" defined    */
                               /* below.                                 */
                               /* ---> I can access mystem:              */
                               /*      Hello                             */

call myproc                    /* Call the label "myproc" defined below. */
                               /* ---> I can access array:               */
                               /*      I am 1!                           */
                               /*      But not mystem:                   */
                               /*      MYSTEM.FOO                        */

say myfunc("a string")         /* Invoke the label "myfunc" defined below */
                               /* as a function.                         */
                               /* ---> I received a string               */

idx = arrayfunc("New item")    /* Invoke the function "arrayfunc" defined */
                               /* below and store the return value.      */
say array.idx                  /* ---> New item                          */

idx = arrpush(array, "Pushed") /* Invoke the function "arrpush" defined  */
                               /* below.                                 */
                               /* We use the fact that REXX will treat   */
                               /* uninitialized variables as uppercase   */
                               /* strings to mock the use of a stem as a */
                               /* function argument.                     */
say array.idx                  /* ---> Pushed                            */

call arrcopy array, new_array  /* Call the label "arrcopy" defined       */
                               /* below, to copy the contents of array to */
                               /* new_array. Note that there's no need to */
                               /* first declare the target stem, it will  */
```

```
                                /* be initialized by our function if it    */
                                /* doesn't exist.                          */

do i = 1 to new_array.0         /* Here we loop over the contents of our   */
  say "New: "new_array.i        /* new_array stem and print the values.    */
end                             /* ---> New: I am 1!                       */
                                /*      New: I am 2!                       */
                                /*      New: I am 3!                       */
                                /*      New: I am 4!                       */
                                /*      New: I am 5!                       */
                                /*      New: New item                      */
                                /*      New: Pushed                        */



exit                            /* Labels can also be used to declare      */
                                /* subroutines and procedures, but since   */
                                /* we don't want to execute them within    */
                                /* the normal program flow, we must EXIT   */
                                /* the program flow before declaring them, */
                                /* or the interpreter will execute them    */
                                /* just like with "my_label" above.        */


/* ===================================================================== */
/*                    subroutine: A label as a subroutine                */
/* ===================================================================== */
subroutine:                     /* This label can be described as a        */
                                /* subroutine and will have access to the  */
                                /* full symbol scope of the script.        */
  say "I can access mystem:"
  say mystem.foo
return                          /* Return to the call statement            */


/* ===================================================================== */
/*                    myproc: A locally scoped procedure                 */
/* ===================================================================== */
myproc:                         /* This label will be treated as a         */
                                /* procedure, a subroutine with a local    */
                                /* scope that can be augmented using the   */
                                /* EXPOSE keyword, to expose arbitrary     */
                                /* values from the global scope.           */

  procedure expose array.       /* PROCEDURE tells REXX that we want a     */
                                /* local scope. EXPOSE is used to "expose" */
                                /* desired values from the global scope,   */
                                /* which lets us access them from our      */
                                /* local scope.                            */

  say "I can access array:"
  say array.1                   /* ---> I am 1!                            */
  say "But not mystem:"
  say mystem.foo                /* ---> MYSTEM.FOO                         */
return

/* ===================================================================== */
/*                        myfunc: Returning a value                      */
/* ===================================================================== */
myfunc:                         /* To create a function, we can accept     */
                                /* arguments using PARSE ARG and return    */
                                /* a value to the caller.                  */
  parse arg str_a
  new_str = "I received "str_a
return new_str
```

```
/* ======================================================================== */
/*    arrayfunc: Add a value to a stem and return the new item count      */
/* ======================================================================== */
arrayfunc:                      /* Let's create a locally scoped function. */

  procedure expose array.       /* Because stems are a kind of sugar for   */
                                /* working with the symbol table, we       */
                                /* cannot pass a stem as an argument to a  */
                                /* function. To be able to work with it,   */
                                /* we must access it in the global symbol  */
                                /* table just like any other variable.     */

  parse arg new_val             /* Accept one argument                     */

  new_count = array.0 + 1       /* Calculate our new item count.           */

  array.new_count = new_val     /* Add our new value to the array.         */

  array.0 = new_count           /* Update the item count.                  */

return new_count                /* Return the new item count.              */


/* ======================================================================== */
/*                arrpush: Pushing an item to a stem-as-array              */
/* ======================================================================== */
arrpush:                        /* Now things are getting interesting!     */
                                /* In this function with access to the     */
                                /* global scope, we accept the name of     */
                                /* an arbitrary stem intended for use as   */
                                /* an array, and push values onto it using */
                                /* the INTERPRET instruction.              */

  parse arg stem_name, push_val     /* Our arguments: the stem's name and */
                                    /* the value to push.                 */

  interpret stem_name".0 = "stem_name".0 + 1"  /* Assume the item count   */
                                               /* is in tail 0 and        */
                                               /* increment it.           */

  interpret "new_count = "stem_name".0"  /* Store the new item count in a */
                                         /* variable for use below.       */

  interpret stem_name"."new_count" = '"push_val"'"  /* Assign a value to  */
                                                    /* the new item.      */

return new_count                /* Return the new item count.              */


/* ======================================================================== */
/*                 arrcopy: Create a copy of a stem-as-array               */
/* ======================================================================== */
arrcopy:                        /* To copy an arbitrary array stem, we     */
                                /* create a subroutine that accepts two    */
                                /* stem names, the source and the target.  */

  parse arg arr_source, arr_target        /* Our args, source and target. */

  interpret "drop "arr_target"."             /* Drop the target to ensure */
                                             /* it's empty. If it doesn't */
                                             /* exist, this will not      */
                                             /* cause an error.           */

  interpret "source_count = "arr_source".0"  /* Assume the item count is  */
                                             /* in tail 0 and store it in */
                                             /* a variable for use below. */
```

```
  do i = 1 to source_count                       /* Loop over the source. */

    interpret arr_target"."i" = "arr_source"."i  /* Copy the value from   */
                                                 /* source to target.     */
  end

  interpret arr_target".0 = "source_count        /* Set the target's item */
                                                 /* counter.              */
return

/* ========================== End of tutorial. ========================== */
```

Floppy Disks and You

==========
Floppy Disks and You - by: Ivan Sorensen
==========

Floppy disks and you
To get anything out of a computer you of course need to software into it first and the default way of doing this has always been the floppy disk: Every Amiga comes with one and when Commodore was around it was the way software was distributed, even if you intended to run it off a hard drive.

For a while, it seemed like floppies were becoming obsolete: CD ROM became the way for magazines to distribute huge collections of software and the internet allowed things to be downloaded directly on to your machine. With the rise of WHDLoad, you could even forego floppy disks completely for daily games playing.

However, in the modern age it seems that the floppy has made a sort of return. A lot of new Amiga games are coming out and while many are intended to just be installed directly, some are distributed in ADF format and while there are ways to mount disk images to RAD, this isnt practical for multiple disk games.

So after many years of neglecting those old drives, we have to dust them off again and that raises the question of replacing them. I stuck with my old drive for years out of habit, but eventually gave in and got a Gotek floppy drive emulator. There are other versions of these out there, but the Gotek is the one I have experience with.

What it does is plug into the Amiga using the normal floppy cable. You then load disk images on to a USB stick and plug it into the drive. The Amiga then reads the ADF exactly as if it was a real disk.

The USB stick can hold multiple files, allowing you to swap disks by clicking the physical buttons on the front of the drive. You do have to memorize or write down what ADFs you have on the stick, but it is rather convenient to work with. It is also quiet and you donâ€™t have to worry about whether a copy protection scheme is going to strangle your disk drive.

As far as the Amiga is concerned, it is a genuine disk running off of a genuine drive so compatibility is very high. I am sure that some edge case must exist, but so far it has run everything I have tried to load on it. Do note that since it is accurate in terms of emulation, that also means speed: The game loads in the time it would take it to actually load from disk. If you are used to hard drive games and WHDLoad, it can be a bit of a crude reminder.

On the physical side of things there are some different configurations you can get. The standard Gotek version seems to be built to fit in a PC style case and will NOT fit smoothly in an Amiga wedge case: You will need to modify your case or trail the cable out the back of the machine and use the drive on its own. I have seen modified Goteks specifically for Amiga usage though. I assume if you have a big box machine, you will be fine.

Other than the difficulty with the physical case (I eventually resolved to just leave my case slightly open, since I use an external keyboard anyway) installation was simple: Just remove the old drive and plug the cables right into the new one. There is no software needed on the Amiga: As far as the computer is concerned, it really is just a floppy drive. The one I purchased came with Amiga specific firmware already loaded, but

these devices are highly configurable.

Are there any reasons to retain your original drives? I am hard pressed to think of one other than authenticity and the desire to keep a given machine in original configuration (or if your hobby has a strict only Commodore era hardware philosophy).
For the average modern user, the Gotek not only is more reliable (and quiet!) than an old drive, it also makes it much easier to actually USE your machine. As anyone who have bought an Amiga 500 can attest to: Its all fun and games until you realize you have no way to actually get software on to your new machine. The Gotek solves that problem: You can simply download disk images from the internet, put them on a cheap USB stick and be up and running on your real hardware.

If you have held off, I highly encourage you to give floppy drive emulators a look. I think you will be pleasantly surprised at how convenient they really are.

A1200 Internal FPU Installation

==========
Internal FPU on A1200 - by: Chris Edwards & Timo Paul
==========

Chris Edwards Presents
If you have one of the Terrible Fire accelerator boards for your Amiga, you already know how great they are. However there is one thing that is not a deal breaker but can be a little bit of an issue for users, and that is the lack of an FPU. Now before you start shouting at me that the majority of Amiga software does not require an FPU and it is a waste of time and effort, let me explain a bit more. For the average user, yes, an FPU is generally not needed; if you play games, write music or do pixel art, then the FPU is simply not an issue.

However, if you are a power user and by that I mean a person who does 3D rendering or image processing then an FPU can range from being helpful to essential. An example are the TF12060 users who want to run LightWave 4.0, you have the power of the LC060 but LW4.0 requires an Amiga with an FPU to run, no FPU no LW4.0.

Sadly there is no space on the board to add an FPU so the ever creative Chris Edwards post peaked my interest. He has posted a very useful video detailing how to install an FPU directly to the A1200 motherboard allowing you to use your, for example, TF1260 with an onboard FPU. I contacted Chris and he kindly agreed to let WhatIFF? publish his video in text form. What follows is a rough transcript of his video.

On the A1200 motherboard there is a blank spot in U0, this is the missing FPU slot, you have two options 1) solder the FPU directly to the motherboard or 2) attach a PLCC IC socket to the motherboard. Chris chose option 2. To start with the sockets are still readily available online, but the FPUs ... not as much as they once were, so you might need to search a bit online. Once you have your socket and FPU, the first thing you will need to do is pop out the wafer in the middle of the socket.

But ...
At this point, Chris mentions that others might object since the A1200 motherboard does not have a crystal to control the speed of the FPU. Here he points out that the FPU is on the motherboard as such it is controlled by the onboard EC020 and runs at 14mhz, thus the FPU will run at 14mhz. Now it is not going to be as fast as your accelerator board but this is not the objective of the task, what we are looking to do is give the A1200 FPU functionality so that it can run software that would otherwise not run without the FPU.

Before beginning to solder he recommends visiting www.amigapcb.org to help with circuit tracing, you do need to know your motherboard revision though. His motherboard is a 1D1 the most common board, watching the video it seemed a relatively straightforward task however, if you have not done any soldering beforehand you might want to get someone else with more experience to add the socket. It took Chris about 30 minutes from start to finish. He proceeded to do some software checks and incredibly the FPU was recognised alongside the TF1230. When he tried to run software that requires an FPU there were no issues, the TF1230 was used as the main CPU and the onboard FPU was recognised allowing FPU-specific software to run. His video can be found here:

https://youtu.be/pUtf2D9IAco

Overall it is a very interesting video and something that if I had more experience with soldering would try. Chris does recommend doing some practice soldering on old hardware but compared to some of the other

soldering I have seen, this one is not too difficult looking. As it stands I will still be asking my local Amiga hardware guy to add the socket for me, I have not had much luck with soldering in the past and do not want to damage my A1200. It is good to know that this is possible and relatively easy and cheap, if you try this yourself let us know how you got along and tell us what you were using the FPU for.

articles

==========

- Articles -

Little Things That Made The Amiga Great

A deep dive into the Amiga OS by Carl Svensson about what components made the Amiga OS stand out among the crowd.

Authenticity, the Amiga and you

Ivan Sorensen looks at the authenticity of using real hardware and with the plethora of upgrades available, asks is the Amiga still an original Amiga experience?

Retro Computer Meetup - Tokyo

The start of a new series looking at local retro computer events from around the world. This time we start with the Retro Computer Meetup in Tokyo and meet a few Japan based Amiga users.

#Articles - WhatIFF? Issue 1.06 - August 2022#

Little Things That Made The Amiga Great

==========
Little Things That Made The Amiga Great - by: Carl Svensson
==========

Introduction

To most people, the Amiga is probably synonymous with the Amiga 500 and that machine was, above all, a video game: something into which you plonked 3.5" floppy disks that transported you to strange, wonderful worlds and hours of fun. To this group, the gamers, the operating system wasn't important. They probably booted Workbench 1.3 only once, took a good look at the garish blue and orange interface, and decided that enough was enough, before going back to blasting enemies in Silkworm.

And why not? The machine was great for games. There are tons of great texts about the groundbreaking audio and video capabilities of the Amiga.

This text is not one of those.

This text is about the later versions of AmigaOS produced by Commodore, from 2.0 to 3.1. While these newer versions (with majors released in 1990 and 1992, respectively) certainly carried the Amiga heritage with pride and were - mostly - backwards compatible, the advent of AmigaOS 2.0 marked a paradigm shift. The shift is best symbolized by the fact that AmigaBASIC - the ubiquitous and mandatory language of any self-respecting eighties home computer - was ditched and replaced by ARexx, a professional scripting language without any kind of graphics or sound features whatsoever.

This paradigm shift is also why so many of us who had the good fortune to come across those versions clung to the Amiga so desperately. A lot of us stuck with it well into the late nineties, years after Commodore's demise. And we were fierce: if you think Apple fanboys are annoying today, be very happy you didn't meet an Amiga zealot in 1995. Things got personal quickly. It was downright ugly.

I still think there were some grounds for our fanaticism, though.

In a time when home PC:s were single tasking DOS boxes with 8-character file names and Ataris and Macs were single tasking GUI boxes, hampering any hacker with their glaring lack of a CLI, the Amiga was a champion of both worlds: It combined the CLI and GUI, leveraging both their strengths. But there was more to it than that, something that's hard to convey in so many words.

Many a time have I come across exasperated Amiga users who, on some online forum, try to explain the greatness of the Amiga's operating system in a few short sentences. It often, quite understandably, boils down to something akin to "It's like UNIX", which I always think is unfair to both of the systems.

Sure, it was an early multitasking system with a capable command line. At the same time, it was also a mature and coherent desktop environment with levels of user-friendliness much more like today's MacOS X than any traditional UNIX. But, contrary to today's MacOS, it never mollycoddled the end user. It opened the machine up, rather than closing certain parts off.

Unlike UNIX, AmigaOS is a system designed for home computer use, lacking a lot of what makes UNIX so popular: memory protection, multi-user support, source code portability and built-in networking

support.

They're very much alike in one aspect, though. Like UNIX, the Amiga's operating system is a collection of a lot of little strokes of genius that come together to make it bigger than the sum of its parts.

I invite you to explore this ingenuity with me. But first, let's get the terminology sorted.

Jargon buster
The Amiga's operating system has been called a lot of things by a lot of people because it consists of many components that different types of users interacted more or less with. Let me try to clear things up a bit:

AmigaOS
Commodore didn't call the operating system as a whole "AmigaOS" until the release of version 3.1 in 1994, the year they went defunct. And even then, it was mostly printed on the box it came in - nothing in the OS calls itself AmigaOS. It's still a useful umbrella term for all of the components below.

Kickstart
This is basically the firmware of the Amiga, residing in the ROM chip. On version 2.0 and up, the ROM chip was 512 kB large and contained not only code for self-tests and bootstrapping but also quite sizable portions of the other parts of AmigaOS.

AmigaDOS
This is, roughly speaking, the command line interface, device drivers, file system management and so on. It was originally based on TRIPOS, a heritage that still shows to this day. (There are newer versions of AmigaOS than 3.1, but this text will only cover the ones released by Commodore.)

Intuition
Best likened to a modern desktop compositor, Intuition handles screen updates and manages windows. The custom sound and video chips in the Amiga had mostly female names (Paula, Denise and later Alice), so I suppose Intuition is a fitting name for an abstraction layer that lets the OS interface with them.

Workbench
This is often used interchangeably with "AmigaOS" but it's really the name of just the desktop environment - which is a workbench, not a desktop: Programs are called tools, folders a called drawers and data files are called projects.

Other terminology

Requesters
Not really a major part of the OS per se, a requester is the Amiga's name for a dialog box. A "file requester" is a file selection dialog, for example. I still often use this term to describe any dialog on any platform, much to the confusion of friends and colleagues. Old habits die hard.

Devices
Devices, much like in UNIX, usually represent peripherals attached to the computer, such as storage media, printers, etc. The Amiga has a number of built-in devices that are automatically mounted, for example the internal floppy drive, DF0: and the serial and parallel ports, SER: and PAR: respectively. Other common devices are DF1:, the first external floppy drive, DH0:, the common device name given to the bootable hard drive partition, and CON:, the text console. There are also devices that can't be mounted but only be accessed using programming languages, such as timer.device which is used to

interface with hardware clocks.

Volumes
Mounted devices (such as DF0, DF1, DH0) can contain a volume. In the case of DH0:, a bootable hard drive partition, the volume is fixed and is represented by the name of the partition, such as System: or Boot:. If you insert the Deluxe Paint program disk into the internal floppy drive, the current volume in device DF0: will be DPaint:.

The RAM Disk
The Amiga always has a RAM disk ready for use and the device name is always RAM:. No mounting, no third party software - it just sits there, waiting. That's clever in its own right, but the really ingenious part about it is that it's dynamically allocated. That means the size of the RAM disk corresponds to the amount of free memory currently available on the system, and that it never takes up more memory than is needed to accommodate the files currently stored on it.

A RAM disk might sound esoteric today, but in the days of slow mechanical hard drives and floppy disks, it was a godsend for unpacking archives, running test compiles and storing temporary working documents.

The downside is that every time you reboot your computer, all the contents of the RAM disk are lost. Which brings us to...

The RAD Disk
RAD, which (somehow) stands for Recoverable Ram Disk, is another testament to the flexibility and ingenuity of AmigaOS. The RAD disk is a fixed size RAM disk that can not only survive reboots but also itself be booted from. It can be used as a small but extremely fast mini hard drive, or for extracting disk images to, for checking out their contents.

Proportional scroll bars
A proportional scroll bar is a scroll bar that grows and shrinks to reflect the size of the scrollable content it represents:

Highlighting a feature like this might seem like a silly point to make, because today all scroll bars are proportional. Not so in the bad old days: Microsoft didn't implement them until Windows 95 and Apple, as far as I know, didn't make the move until MacOS 9 in 1999. In a time when computer mice had no scroll wheels, this was a nifty feature indeed.

The Text Editors
AmigaOS shipped with three different text editors: Ed, Edit and MEmacs. The latter is a port of, you guessed it, Micro Emacs, and will of course appeal to Emacs fans. The other two are sort of curiously named: Edit is a line editor mostly resembling ed on UNIX. The Amiga's Ed, on the other hand, is a full screen editor with both mouse support and an extensive command language. Amiga Ed is probably best likened with vi.

All three of them are capable and powerful editors. Both MEmacs and Ed make use of the Amiga's drop down GUI menus and feature support for macros and extensive configuration.

Ed's command mode is, as mentioned earlier, somewhat reminiscent of vi. Consider the following examples:

Change the next three occurrences of "data" to "gubbe":
3 E/data/gubbe/

Move to top of file, repeat (RP) replace of "data" with "gubbe" until end of file:
T; RP E/data/gubbe/

Move to top of file, find all occurrences of "gubbe" and insert three blank lines after every occurrence, repeat until end of file:
T; RP (F /gubbe/; 3A//)

Ed also has something called an ARexx port, which completely intentionally leads us to...

ARexx
ARexx is an improved and Amiga-ized version of IBM's REXX language. It's a capable (albeit somewhat dated, these days) general-purpose scripting language with excellent functions for easy string parsing.

Through an interprocess communication interface, it can also act as "glue" between various programs with so-called ARexx ports. For example, if both your text editor and Desktop Publishing package have ARexx ports, you can write a script for automatically inserting the text you've just written in the editor into the desktop publishing program by interfacing with them both using ARexx.

Ed, the editor described above, exposes its command language through its ARexx port, allowing users to create highly advanced text processing scripts.

No CD
While AmigaDOS does have a CD command, it's usually not needed. Any command that evaluates to a valid path will automatically change the current working directory to that path:

```
1.System:> Work:Code/ARexx/WebServer
1.Work:Code/ARexx/WebServer> _
```

File Systems
OFS (Old File System) and FFS (Fast File System) are the Amiga's native file systems. Thanks to the modular and ingenious nature of the OS, the Amiga can, much like UNIX, handle an arbitrary amount of file systems. Version 3.0 of the OS shipped with CrossDOS, a utility to mount MS-DOS FAT volumes - but a lot of others have popped up as well, from third-party native replacements for FFS to ports of for example EXT2.

Assigns
An assign is a kind of fake logical volume that can point to any path. It's great for deep directory structures. Say you've got a current programming project going in Work:Code/ARexx/WebServer. That's quite a handful to type often. By creating an assign,
ASSIGN wserv: Work:Code/ARexx/WebServer, you can now access files and CD to your project directory by referring to wserv: instead.

An assign can also span several physical volumes. The Amiga's standard assign where it looks for dynamic libraries is LIBS:, pointing at the Libraries directory of the current boot volume. But what if you're writing a new such library and don't want to litter your system directory with your development versions of it while testing?

Not to worry, you can ADD to any assign:

```
1.System:> Assign LIBS: Work:Code/C/HTMLParseLibrary ADD
```

AmigaOS will now look for dynamic library files in both paths!

Libraries
Much like libraries in UNIX, this is a way of sharing common functions between different programs. Amiga libraries are always dynamically linked and yet, for some reason, there never seems to be any problems with backwards compatibility. For an end user, things just generally work.

Commodore shipped AmigaOS with libraries for everything from mathematical functions to something called ASL, used for creating file and font requesters.

ASL
The simplest way for a programmer to ask the user to pick a file or a font is through the ASL library. Using a standardized API, a requester (dialog) for saving or loading a document can be presented to the user. The dialogs have functions for pattern matching and keyboard navigation.

Sadly, it doesn't stand for "Age, Sex, Location" but rather "Amiga Standard Library."

Datatypes
Datatypes are, in a sense, a bit like dynamic libraries for file formats. Instead of relying solely on the capabilities of, say, a graphics program to implement loaders for every conceivable file format if it supports Datatypes, it can load images of any format for which a Datatype is installed in the system.

This is a great way of future-proofing software, ensuring that even older programs can load and display newer image formats. When the PNG format was created in 1996, many Amiga programs made years earlier could load those images through the use of a third-party PNG datatype released as freeware.

ReadArgs
While AmigaOS doesn't have online documentation like UNIX man pages, it does have a standardized way for CLI programs to read arguments. This is facilitated by the ReadArgs function, designed to help programmers implement this way of reading arguments.

The ? (question mark) argument always produces a list of a command's arguments and their kind, and then awaits user input of said arguments:

```
1.System:> Delete ?
FILE/M/A,ALL/S,QUIET/S,FORCE/S: _
```

So what does this mean? It means that FILE can be multiple strings (/M) and that it's a required argument (/A); ALL, QUIET and FORCE, meanwhile, are boolean flags, or switches (/S).

There are also patterns for keywords (/K), numbers (/N), and a handful more. Once you get the hang of it, it's a very helpful reference system for refreshing your memory, and the argument names tend to have curiously sensible names.

ReadArgs is refreshingly easy to implement in shell scripts, ensuring conformity to standards even for the simplest of tasks.

AmigaGuide
AmigaGuide is an early hypertext format, originally used by Commodore for their official developer documentation, but delivered as a standard component of AmigaOS 3.0 and up. It's simple compared to modern HTML: it can't display inline images or play background sounds, but you can create links to arbitrary files that are then displayed separately.

It quickly became popular and before long, most freeware, shareware and public domain programs came with AmigaGuide documentation.

Installer
In the beginning, hard drive installation of programs was made using shell scripts. Seeing the havoc wreaked by such scripts, Commodore eventually created a standardized software installer. It's a brilliant

little tool for selecting installation paths, creating directories, comparing file versions, copying files and modifying startup scripts.

Pretty much all developers adopted this right away, including PD/Freeware authors, much to the enjoyment of end-users. To this day, installers for my crappy Amiga software remain the only really meaningful thing I've coded in a purely functional language: Installer scripts are written entirely in LISP.

Because of the powerful nature of having a LISP dialect and easy access to a subset of the Amiga's GUI widgets, the Installer has also been used to create simple adventure games.

Standardized File Structure
Every Amiga installation looks basically the same, especially where it really matters. We've already covered shared libraries and their residence in LIBS:. The same pattern is used for the rest of the system files: Device drivers are always in DEVS:, file system drivers and other low-level stuff in L:, AmigaDOS commands belong in C:, shell scripts (including the bootup scripts, Startup-Sequence and User-Startup) live in S:, typefaces are always in FONTS:, and so on.

Resident programs
Some programs, most of them AmigaDOS commands, can be made resident - that is, stored in memory for future use. This is useful to avoid excess disk access when performing certain tasks.

RequestChoice
AmigaDOS has a simple command for asking the user a Yes/No question, called Ask. In version 3.0, Commodore introduced the commands RequestChoice, for making a multiple-choice GUI requester and RequestFile, for opening a file requester. A nice example of how integrated the GUI and CLI environments are on AmigaOS.

.info Files
A bit like with MacOS X, if you give a disk to an enthusiastic Amiga novice and ask to have a few files copied onto it, you'll probably find it littered with strange files that serve no apparent purpose. They always have the same name as another file, but end with ".info".

These are the Amiga's icon files, containing not only the actual icon image, but also a bit of metadata. For a data file, that can be what program to open the file with, which means that if you double click a picture created with, say, Deluxe Paint, then Deluxe Paint will be started and then load the picture in question. This is called the file's "default tool".

The .info files also contained positional information: the desired coordinates of the icon within its parent window. For drawer (directory) icons, it also stored the size and position of the window opened to display their contents.

Other noteworthy metadata is represented by the Tool Types, which can be described as settings or arguments for a program. For example, such a tool type can be used to control if Deluxe Paint should display the coordinates for the mouse pointer's current position.

A system with stand-alone icon files might seem odd today, but instead of having centralized icon storage or embedding the icons into the binaries, this method meant that tooltypes, default programs and positional information could easily be transferred between systems. It also gave the users freedom to modify or even draw new program icons or delete them completely if they wanted to save disk space.

Several beautiful icon sets were released as third-party shareware, most notably Magic Workbench.

Closing words

While certainly not a UNIX clone, the Amiga was still a great preparatory platform for tackling other operating systems further on. When I first stumbled across Linux (and other *nixes) in 1995, a lot of stuff instinctively made sense to me. I easily drew parallels to AmigaOS and was already familiar with a lot of the concepts: devices, mounting, task priorities, shell scripts and scripting languages, Emacs and much more. And as much as I enjoy UNIX today, it's when I sit down in front of an Amiga that I truly feel at home and in control.

I hope this text has given you some insight into what made the Amiga great. There are of course many more things, but these are a few that have stayed with me. Some concepts, like Datatypes, are still unique to the Amiga and would be an interesting and useful addition to an operating system even today.

And remember - Only Amiga makes it possible!

Authenticity, the Amiga and you

==========
Authenticity, the Amiga and You - by: Ivan Sorensen
==========

Authenticity, the Amiga and you
Whenever retro-enthusiasts are discussing ways to interact with old software, you inevitably get the discussion about what is authentic: Is emulation authentic? Is a Raspberry Pi in an Amiga case authentic? What about Morph OS or OS4? What about an FPGA system? Surely, an actual Amiga is authentic right?

Now first and foremost I want to preface this by saying that authenticity ultimately depends on what specific things you value and whether it is important at all is ultimately up to the end user. The Amiga fan base has people who love the hardware and people who love the software and they are not always the same people.

But that doesn't mean we can't stop and take stock of the situation and maybe try to explore the topic a little bit. Maybe we can then all get a better understand of each other and even ourselves?

On the surface it appears to be simple enough: An Amiga 1200 is authentic (because it is the real deal). An FPGA system that reproduces the Amiga 1200 chips is less authentic and a computer running an emulator is the least authentic. Case closed and we can all go home.

But can we really?

Let's take a closer look at our Amiga 1200.

My Amiga today is fairly modest compared to many systems enthusiasts have built: A 1200 with a fast 68030 card and a few other upgrades. Pretty good but a far cry from the maxed out machines people like to set up. How authentic is that system? Well, the case and board is 1992 vintage so that's very authentic. The ROMs are 3.1 so they are authentic enough.

But the accelerator card was made in 2010 and is faster than any 68030 available at the time. The hard drive is a compact flash card in an IDE adapter, the mouse is a PS2 mouse with an adapter, the joystick is brand new, I've attached a PS2 keyboard using a Lyra and the disk drive is a Gotek.

Do any of those change that this is an authentic system? Well, it must right? All of this is equipment that was not available at the time. But at the same time, none of it really adds capabilities that were impossible. The Gotek is much more convenient than a real floppy drive, but it still acts like a floppy drive. My brand new joystick is much more precise than the Quickshots I got off ebay but it's still just a joystick with 9 pin connector and so forth.

If we had to rate it on a scale from 100 (completely authentic) to 0 (completely inauthentic) maybe its around 70?

It certainly feels authentic when I sit and use it but what was my system like at the time? It was an Amiga 1200 with an external floppy drive and that was it.

For comparison I looked up a Mac from 1992 and my accelerator card has 6 times as much RAM and a much faster CPU. My relatively humble Amiga today would have been a beast of a computer in 1992. So clearly my Amiga now cannot be authentic to the experience I had as a 12 year old.

That is leaving aside the software. Now I can click on a game and through the magic of WHDLoad it loads the game into memory and runs the same way it would on a cartridge based system. Meanwhile in 1992 I would have been swapping disks and waiting for the grinding of the disk drive (while hoping the rumors that copy protection could destroy your drive were false. I never did find out).

Would sitting in the dark room with a stack of floppies and an old ball mouse be more authentic? Yes. Undeniably so. That IS the authentic Amiga experience for me.

But would I want to again? Well, I did but I also upgraded my system when I was able to in order to get more functionality or more convenience or more capabilities.

So I guess my conclusion is that I am happy with 70% authenticity. And I expect I'll be happy at 80% or at 60%. What is your level of authenticity? What did you build your system to do?

Take a moment to think about it the next time you boot it up.

Retro Computer Meetup Tokyo?

==========
Retro Computer Meetup Tokyo - by Timo Paul
==========

In an attempt to broaden the scope of the magazine we will be starting section focusing on local Amiga or retro computer groups/meetups around the world. To start this off we will be looking at the Tokyo Retro Computer Users meet up in Kayabacho, Tokyo. I had been introduced about this meet up by Edoardo or has he is also known as `kinmami` on EAB. Last year he had replaced the CAPS and DAC chip on my A1200, making it sing like new again. A few times he had invited me to attend but due to the covid situation I had been hesitant, however now with my third vaccine well and done, it was time to venture up to Tokyo.

I live in Chiba which is about 50 minutes by train to Tokyo station, getting to Kayabacho was a short trip of only 47 minutes by rapid train on the Keiyo line. On the train I had my A1200 with me again, ready to have the TF1260 firmware updated by Edoardo. It was quite exciting as I had not been out since the covid situation and more so I had not met any Amiga users in Japan since I had moved here so I was very much looking forward to meeting similar minded Amiga junkies.

Upon arriving at Hatchobori station, it was a short walk to the exit that would take me to the Co-Edo Co-working Space Kayabacho, according to Google maps it should only take 7 minutes. The map looked very simple but I seem to have any amazing ability to confuse map directions. Following the map printout (Yes, I do not have a smartphone and still find my way around like it's 2000 again) I ended up on the wrong side of the river, brilliant! Quickly I backtracked to Hatchobori station and started again, however this time I asked some friendly Tokyoites if I was on the right track. Everyone I asked gave me great directions until I was very close and .... still could not find the place, I went to a Lawson convenience store and asked the staff where I could find the Co-Edo building. He looked at me for long moment, then pointed behind me. To my surprise there it was behind me, feeling like an idiot I apologised and thanked him.

Finally arriving at the building, it was like most typical Tokyo buildings, thin and tall, the entrance was cramped but effective. The reception was on the 5th floor, when I got there I told them I was with Sean, paid my Y1,100, filled in my details and received my one day visitor pass. From there it was down to the 4th floor to meet the guys for the first time. Entering the 4th floor, it was refreshingly cool thanks to the air-conditioning. Splitting the room was a smoked glass window with the meeting taking place on the other side. I cautiously opened the door and the first person I met was Edoardo, I introduced myself and then proceeded to meet the rest of the people there. This was around 10:30 am and at that time there were not too many people, so it was easy to meet everyone. Sean is the organiser and you can find him walking around taking photos and videos asking questions about the various computers that people brought in.

The room had two main tables and one of them were some very large computers, one being a Casio FP-1100 brought in by Sean. He kindly showed be around the machine and sat me down for game or two. It was a very striking machine that would not look out of place in any of the current electronic stores. The design was very modern looking, aside from the fact that the keyboard cannot be disconnected. That it itself would be an issue but the keyboard was almost as big as the computer!
Curt brought in a Basic Master Level 3 computer which was another monster of a computer, unfortunately I never got round to having a play

on that one.

The guys there were very friendly and quick to answer any questions I had. If you wanted to try a machine they would you sit down and expliain the machine.

The atmosphere was very relaxed and friendly, I immediately felt at at home there. After getting introduced to the rest of the guys it came time for my TF1260 firmware update, it took a little while longer than expected since Windows had to update, or as Edoardo said "It's plug and pray!", luckily the update went without hitch and then it was on to my TF1260. The firmware was fast, pretty much instant. From there on the Amiga would just sit doing some tasks to check its stability.

A little later some more people arrived, first Michelle with a massive suitcase, upon opening it up she had a NEC PC-8801MA plus 14" CRT monitor, it must have been quite a journey for her lugging all 35kg of that from Yokohama.

However as being someone who does not have the money or space to buy other computers this was a fantastic opportunity to get up close with rare computers and even try the out. Michelle introduced me to one of the most difficult games I have ever played, Warpman, I think I demonstrated my lack of gaming skills quite well during that time. Michelle tried her best to guide me how to play the game but my hands and brain seemed to be doing the opposite of what I wanted.

Shortly after Michelle arrived Robert another newcomer came in lugging another huge box, this was going to the highlight of the day for many people as when he unwrapped the computer it was a SGI Iris Indigo. Suffice to say many of us had our interest peaked at the sight of that rare beast. The next few hours flew by as we played with the various hardware talked shop, computers and games. It turned out that Robert was another EAB member and was an Amiga user, bringing the total of Amiga guys to three at the meet up. Suffice to say talk quickly moved to the Amiga, it seems no matter in the world if you have Amiga users they are happy as ducks in water to fritter the hours away talking about the Amiga.

Sven was originally working on his Z80 mini PC, a very cute looking computer, but when Robert unveilved his SGI all thoughts of the Z80 were gone. As the day passed on more and more people turned up, unfortunately I did not have the chance to talk to all them. Saburo I am sorry I did not have a chance to chat to, hopefully we can talk properly next time.

Overall it was a great day, meeting people interested in old computers and the chance to see and touch some incredibly rare computers was great fun. The fact that the owners there are happy to tell you about the computer's history, how to use them and answer any questions you may have, makes it a very welcoming and enjoyable event. If you are an Amiga user or retro computer collector and want to spend time with similar minded people then I recommend attending an event. I will be attending the next event at the end of August or early September with my A1200 for another firmware update to be done and maybe have an FPU socket installed onto the motherboard.

If you live Tokyo or Chiba and are an Amiga user I hope we can see you at the next event!

Welcome to voices of the people behind Amiga hardware and software.

```
==============================================================================
                      -  Talking Amiga -
------------------------------------------------------------------------------

                      -=  Edoardo Auteri Interview  =-
   Continuing our focus on Japan this issue, we talk to Edoardo, a
   Tokyo resident servicing the needs of Amiga users around Japan.
                       -=  Pawel Nowak Interview  =-
   The author of Infinity Music Player or IMP, we talk about Pawel's
   history with computers and the Amiga.  Find out more inside!

------------------------------------------------------------------------------
            #Talking Amiga - WhatIFF? Issue 1.06 - August 2022#
```

Edoardo Auteri

Edoardo, thank you for taking the time to answer our questions.

Can you tell us a little about yourself?

Hello, glad to be here. Thank you for the fantastic opportunity.
My name is Edoardo Auteri, European, born in Italy in 1977. I moved to Asia in 2010 and from 2014 working and living in Tokyo. I am fascinated by the eighties of which I keep vivid memories and try to replicate the best aspects in everyday life. I am working in the semiconductor industry since 1997.

How did you get started with computers?

I have always been interested in electronic games and entertainment. When possible, I would go to arcades to play games in the eighties. In 1985 a school mate invited me home for Sunday lunch and afternoon study. His father challenged us to play Zaxxon with the Commodore 128 he just bought. I was impressed with the isometric view, collisions, and sound effects. Also playing with a paddle was a memorable experience as I just used the arcade joysticks. My first home computer was the Commodore 64C given to me by my parents in Christmas 1986. And from that moment on I have not stopped interested in hardware.

What was your first Amiga experience?

My first model was the Amiga 500 purchased in the summer of 1988. Advertisements in newspapers and television channels were rife in Italy. There was a murmur in classrooms about "a new home computer capable of speaking voices and playing orchestral music". I was using the C64 with satisfaction, yet I was curious. I first tried the A500 at friends' and Commodore dealers in late 1987. Japanese consoles, Atari, and Spectrum did not shock me (not yet at least) while I was thrilled by the idea that a new Commodore model could go beyond the potential of the C64.

What attracted you to working with Amiga hardware?

The genius in the design of the custom chips and the flexibility of the Amiga structure represent an immense interest in improving, expanding performance. Furthermore, the possibility of managing audio and video with relative ease achieving results comparable (for the 80s and 90s) to television studios was astounding. Both on C64 and on Amiga I experimented with DIY boards connected to serial, parallel and expansion ports. I wrote high- and low-level language programs. In short, I prepared myself for the future study and work. I would not be in Tokyo, and I would not deal with technology and semiconductors if I had not kept practicing in depth with Commodore Amiga.

What kind of work do you do on the Amiga?

Nowadays I delight in repairing all models. From a simple replacement of worn capacitors and logics to the modification/fixing of old original layout boards. Continuing with the design and prototyping of new expansion cards, adapters, and accessories that I usually share with the community (EAB, Discord, Hackaday and PCBWay). Sometimes I like to load and run classic games of course.

You currently live in Japan, what brought you to Japan?

Japan strongly influenced my Italian childhood. I do not remember anything other than fashions, hobbies, entertainment directly or

indirectly imported from Japan. I am not a fanatic of everything made in Japan. However, I appreciate good manners, respect for one's word, lifestyle. These are aspects that Italy has only partially been able to perpetuate. Before Tokyo I lived and worked in Seoul, Shenzhen, Singapore in the semiconductor market.

Can you describe the current Amiga scene in Japan?

I know few Commodore Amiga enthusiasts and collectors in Japan. These are not usually gents and ladies experimenting with their machines. Friendly people interested in news. They were even former employees of the Japanese branch of Commodore (they showed photos with Jay Miner). There are also friends from the US, UK and Australia living in Japan and using Amigas. They are running YouTube channels in English and Japanese showing the pixel art, and hardware & software tutorials, for example.

You are one-half of the Ramixx500 Amiga 500+ Main board replacement project, can you tell us a little behind the idea?

The project author is Giorgio Moscardi, known as Sukkopera. He is based in Turin, Italy. All credits to Giorgio and his formidable ability and meticulousness. I validated the project by bench evaluation, finding bugs, and suggesting Giorgio improvements. The purpose of the project, well explained on Hackaday, is to improve the original concept to preserve the design and make it usable at reasonable prices. Sordan (Ireland) is the online shop that helps supporting the project. Thanks to Giorgio for his extraordinary effort.

What kind of improvements do you plan to incorporate into the new motherboard?

We have redesigned the power and ground planes. Additionally, we have done a more reasonable selection of the Bill Of Materials list, added video sync buffer, jumpers to switch among internal/external FDD, HDD led activity circuit, ESD protections, etc. We have worked out a new Video Hybrid, also known as VIDIOT, able to output 15KHz colorful composite and S-VIDEO signals for the A500 and A2000, A3000.

You are working on a new motherboard however the board will need original chips, do you think it will be possible to re-create all the custom chips allowing the Amigas to live on for longer in physical form?

In theory anything can be reprogrammed using a proper FPGA. In fact, there are tentative of making a brand-new FATTER AGNUS, DENISE, etc. For example, BUSTER for A2000 is implemented with a FPGA and available to users. The 680xx microprocessor itself is entirely reinvented with FPGA and will be available soon. Dave Haynie is studying an improved version of SUPER BUSTER for the A3000 and A4000. Today the market does suffer the shortage then projects slowed down or stopped. We will see how things will go after 2025, I suppose.

What Amigas do you own? If you have any other types of computers please feel free to mention them.

Various models of A500/+, A600, A1000 (and Phoenix), A2000, A1200, A3000, A4000 (desktop), CD32, CDTV. In addition to C16, Plus4, C64, C128, Sony/Toshiba/Casio MSX, MSX2+, Sharp X68000XVI, Atari 800XL, Atari 1040 STE and vintage consoles from Nintendo, Panasonic, and Sega. I miss Amiga tower versions, but I own all motherboard/hardware desktop revisions, including SAM460ex.

What is your main Amiga setup?

A1200 with TF1260 accelerator, kickstart 3.2, Flash Floppy drive and CF HDD. Definitely 32bit AGA based.

What would your dream Amiga be?

It is nice to see that over the past twelve months they have released (numerically) more quality games and enhanced versions of classics of Amiga than new Sony PS5 releases. There is excitement and there are innate talents working from behind the scenes. I would like to play the Street Fighter II Champion Edition conversion built out of Scorpion Engine actually. I have no dreams of new hardware/accessories.

Stakeholders need new software applications and tools. I hope that the global Commodore / Amiga community will become more supportive, meritocratic, open-minded, and put aside feelings of hatred and greediness.

Thank you for your time Edoardo, do you have any final words or anything you would like to mention?

I appreciate the kind interest and wishing everyone the best possible. Please feel free to contact me further on SNS and discussion forums.

https://jp.linkedin.com/in/edoardo-auteri-mba-35096240/en?trk=people-guest_people_search-card

https://www.pcbway.com/project/member/?bmbno=74CA1DF4-D529-4B

http://eab.abime.net/member.php?u=34470

https://hackaday.io/project/180442-rmixx500-amiga-500-mainboard

Pawel Nowak Interview

Pawel Nowak, thank you for taking the time to answer our questions.

Can you tell us a little about yourself?

I'm Pawel, I live in Poland, specifically in Krakow. Professionally deals with programming. I also have my own small printing company which I also use in the retro world - I publish games in large boxes, CDs or even cassettes. I'm 39 so I'm still very young :-) I have a wonderful wife and two children. I don't have as much time for my hobby as I used to, but I try to support the Amiga market as much as I can. I am also a collector of old computer junk.

How did you get started with computers?

My first computer (when I was 7) was the Atari XE. It is thanks to him that I learned what it means to program. After about two years, I was able to exchange it for the Amige 500, which was a total novelty for me. I had no idea there was such a thing as a computer mouse. What the Amiga offered exceeded all expectations. I stuck to my A500 for a long time, everyone around them had a great A1200 with 030 and 040, and I was still faithful to my A500. Finally my friends convinced me and I bought a used A1200. I could do anything on it. After a while I got Blizzard PPC. This adventure continued for a few more years until my card broke. for about 10 years I lost my interest in Amiga. After this time, I started collecting retro gear, including going back to the amiga and starting coding again.

What was your first Amiga experience?

After replacing Atari with Amige in 1992, I was still a kid, but I already had some knowledge of how to use a computer efficiently and could deepen it further. I was interested in games on the average, one day we bought VBS, and this opened access to huge collections of software, disk magazines and other productions from the demoscene. At that time, in Poland, no one was concerned about copyright. On the one hand, it is bad because it was hard for people to give up piracy later, and on the other hand, it made development possible. I started to get to know new contacts, with whom I exchanged letters, until I got to the right people and entered the scene as a swapper, I started going to an Amiga party and got into it all totally. In the last days of my swapper life I belonged to groups such as Appendix, Nah-Kolor or Mawi.

What attracted you to working with Amiga hardware?

The Motorola 68k is the most attractive software processor I know. You don't need a lot of knowledge to quickly learn its secrets and start writing your own code. Amiga hardware is a great fun environment for both beginners and advanced players. You can do everything your way and enjoy the final result even if the requirements are greater than the goal of the A500 Amiga. Amiga and assembler is the best combination.

What kind of work do you do on the Amiga?

It's quite simple in my case - 90% of the time I sit on IMP (a lot is happening on our Polish channel), 10% watch demos. Sometimes I bend these proportions and check new programs or games.

You live in Poland, can you describe the Amiga scene there?

The Amiga scene in Poland has always been large, both in the times of the amiga's glory, when a lot of demoscene productions were created, as well as now, when the demoscene is no longer active, but various types of meetings, just amiga owners, the larger and smaller ones take place all the time in various corners of our country. I have friends a few hundred km away, and it seems to me that I meet them every now and then, and meeting someone after many years is my everyday life. A lot of people come back to using amiga, despite the fact that in our country, probably everyone is as busy as possible :)

Infinity Music Player (IMP) was your creation, what other Amiga software did make before IMP?

On aminet.net you can find some of my programs, some have been there for over 20 years. From newer stuff, someone may associate me with games that I create in the Project R3D group, such as Tanks Furry, Bridge Strike, Zgr3d or Astro Blox Revisited - all are available for free download on aminet.net or in a physical version (box) on the website www.cdlabel.eu

What gave the idea to create IMP?

I was looking for an idea for a network program. In general, I was interested in how to program the internet in amiga. First, I thought about a game that would use it, for example for online play. But such a game would take years to develop, so I was still looking for new ideas, and finally it dawned on me that having a show collection of modules, it could be easily used and then the first version of the IMP was created, which worked exclusively in CLI, loaded a random module one by one (it additionally buffered them, to play next one right away). The idea was well received in the community and I immediately started adding more stuff, information about samples, preview of patterns played (still in CLI :)), number of online users, simple chat etc, until it turned out that the GUI version was inevitable.

There must have been many hurdles during the project, what were some of the most difficult ones?

The hardest part was bsdsocket.library, which was news to me, and there is not much documentation for the assembler. Something that worked on the winuae wasn't working on a real amiga, then something that worked on miami didn't work elsewhere, something that already worked on every tcp / ip stack suddenly had problems with different drivers - especially atypical NICs for large amigos. Modules player clashed with interrupts, etc. In the end, all the problems were solved - thanks to the people who tested the program (because I do not have access to every possible configuration myself at home)

The software has many features already, do you plan to improve or add to the program?

At this stage, there is no greater need for software development, but sometimes you suddenly get an idea and implement it. I don't think too much about it anymore. However, I am developing an additional thing that will soon see the light of day. It is an impbox - an esp32 based device plugged into a parallel port in the amiga. Thanks to it, the imp does not need an Internet connection, the wifi module in the impbox is used to communicate with the server, so the strobe configuration on the Amiga side is omitted, more memory remains in the amiga itself. Impbox is equipped with a display with information about logged in users, vumeter. It is still a development issue, but it works quite steadily. the basic cdtv amiga achieved a transfer of around 70kb/s, which beats the plipbox you know at the moment.

What Amigas do you own?

Currently I have 36 amiga, and a lot of other computers and consoles. There would be no space to describe it, but when someone visits me, you can always count on a little tour and getting to know my collection. An almost up-to-date list can be found here http://szajb.us

What is your main Amiga setup?

I have used many amiga interchangeably. The most common are amiga with warp1260, amiga 4000 with 060 and zz9k (with additional sound card), amiga 2000 with tf536 and amiga 500 with aca1233n. I also turn on my cdtv from 010 a lot to test the impa as well. In the second apartment I have a600 with v2 which I use only for imp and testing the impbox.

What would your dream Amiga be?

I haven't looked at it in such terms for a long time. Amiga with aga and fast 060 is all that is enough for me to demoscene. Older productions can be run on any other. I miss a4k mediator a bit to take advantage of the PPC (bigfoot) that lies and waits in the desk drawer, and to have access to the rest of the ppc demos. Here, the equipment with morphos comes to the rescue, which fill me up a bit. I can enjoy any amiga with no matter what configuration and I don't need anything else.

Thank you for your time Pawel, do you have any final words or anything you would like to mention?

Thank you that you are still with the amiga, for me it is a huge chunk of my life and something that shaped me in its own way, I cannot imagine my life without it now. Thank you all for your great support, turning a blind eye to various inconveniences in my productions, for healthy criticism and for the fact that we have something to talk about.

Thank you for inviting me to this interview, it's great that such magazines are still coming out in the Amiga media world.

Welcome to the links section!

==========
Links
==========

www.amiganews.net/en/
One of the most up to date and comprehensive sources for
Amiga news from around the globe.

www.eab.abime.net
English Amiga Board, the majority of the users are from
Europe, a very friendly and helpful Amiga forum.

www.exxoshost.co.uk
The resource if you need help or information about the
new Terrible Fire cards.

www.amibay.com*
A great place for buying and selling Amiga hardware,
they also have some very nice people to give advice about.
Of recent it has become very hard to register on the
website since their server migration.

www.amiga.org
Probably the longest running Amiga forum on the world
wide web, most American Amiga users are on this board.

www.amigafuture.de
The longest running Amiga print magazine, issues are
available in English and German.

www.amiga-addict.com
Newest English Amiga print magazine, if you are missing
the likes of Amiga Power and Amiga Format, then this is
the magazine for you.

www.ka-plus.pl
Commodore and Amiga print magazine avilable in English
and Polish. Lots of games content.

==========

If you have a link that you would like added to list
contact: submissions@whatiff.info

Welcome to the Amiga vendors section!

```
==============================================================================
       Amiga Vendor Links
==============================================================================

    Alinea
        https://www.amiga-shop.net/

          Amedia Computers
      https://www.amedia-computer.com

      Amiga 68k
   http://amiga68k.com

          Amigakit
        https://amigakit.amiga.store/

          Amiga MC1
   http://amiga1.com

                        Amiga Shop
   https://www.amigashop.org

   AmigastoreEU
          https://amigastore.eu/

   Amiga on the Lake
   http://amigaonthelake.com/

   Arananet Retro Products
        https://www.arananet.net/pedidos/

          Arcade Shopper
         https://www.arcadeshopper.com

         BuyElbox
          http://buy.elbox.com

   Electronics Is Fun
         https://electronicsisfun.com/

   GGS Data
   http://www.ggsdata.se

   My Amiga Shop
   https://shop.myamigashop.com

   Relec
   http://www.relec.ch/fr/

   RetroAMI
   https://retroami.com.pl/

   Retro Passion
                        https://www.retropassion.co.uk/

   Retro Ready
   https://retroready.one/

   Sell My Retro
   https://www.sellmyretro.com/

   Sordan
```

https://sordan.ie

Tindie
https://www.tindie.com/

Vesalia
https://www.vesalia.de/

#Amiga Vendor Links - WhatIFF? Issue 1.06 - August 2022#

Ivent's Corner

==========
Ivent's Corner - by: Kevin Saunders
==========

In recent times I have had more interest in developing for the Amiga Vampire platform. Over the years I have followed it with great interest. So far we have 3 games, *Apollo Crown*, *Apollo Blocks* and a Tech Demo for *Apollo Menace*. Gunnar and his son Arne are the main coders I work with.

We are currently working on *Apollo Space Invaders* and this is a great fun game to play and is evolving into something better.

We have plans on a few new games *Joust* (Arcade conversion), *Arkanoid* clone, *Mutant Monty* remake (Amstrad CPC conversion. I have made all the maps for both games and focusing now on the redrawing of various elements such as the enemies or menu design.

The big news though is a brand new game, exclusive for V2 and V4 card Vampire machines is a vertical shoot-em-up called *RED DWARF STAR*, some fans may have noticed the posting recently. I decided to simplify the title as some people understandably got confused with another shooter I am proud to be part of with Richard Loewenstein *Reshoot Proxima III* we are getting ever closer to the release of this game (later this year). Richard, I'm sure this will be our best version yet, impressive coding, it seems almost magical. Technically speaking I'll be pushing every part of the hardware. The main difference between the V2 and V4 is the 16bit sound for all V4.

If you would like to see from a day-to-day perspective what I work on please come visit my own Discord channel. Post feedback on games, suggest games to port, or just hang around in the background. I would say this is the best way to see up-to-date changes if you're interested.

https://discord.gg/K3f7UyYDT9

Discord is my main communication channel between all the developers I work with and it's far quicker to post a quick update.

Other News:
If you have not seen the latest info on the Apollo range of Accelerators, the team have just announced the Manticore V4+ Accelerator for the A600. For more info please visit here:

http://www.apollo-core.com/knowledge.php?b=1&note=38662

Other games:

International Karate Masters

Another game that is quietly being worked on for V2, and V4 machines is a game called *International Karate Masters*, I'll be helping with background scenery and possible pixel cleanups of the characters, more info coming soon. I've been a big fan of IK+ on the Amiga and am excited to be part of this.

Amiga Forever.
Invent | Kevin Saunders

What's coming up in the next issue?

```
==============================================================================
   Next Issue - 1.07 October 2022
==============================================================================

  Thank you for reading issue 6 of our little magazine, for our next issue
  we will have some more Aminet reviews of BootPicture, AmiX550 BBS, Wet
  Weather and more.  In addition to the reviews you can expect the
  tutorials for Brilliance and LightWave to continue along with articles
  John Scolieri, Ivan Sorensen and Kevin Saunders.  For issue 7 we will
  have an interview with Harrison the founder of AmiBay and Kipper2K plus
  a lot more, see you in October!
------------------------------------------------------------------------------
            #Next Issue - WhatIFF? Issue 1.06 - August 2022#
```

Meet the writing staff of WhatIFF?

==========
Writing Staff
==========
WhatIFF? is an independent magazine created by Amiga enthusiasts from around the world looking for a more serious magazine. The main writers at the moment are a small group but hope to grow in coming issues.

* Main Writing Staff *

Timo Paul Steve Lord Kevin Saunders

John Scolieri Ivan Sorensen

* Guest Writers *

Maddi Carl Svensson Daniel Reimann

We are always interested in new articles, guides or stories, if you have something you wish to submit please send it to:
submissions@whatiff.info

#Writing Staff - WhatIFF? Issue 1.06 - August 2022#

timopaul

Name: Timo Paul

Aliases: amifan (EAB)

Contact info: timopaul@whatiff.info

Websites: www.whatiff.info
  www.binarybeats.me

stevelord

Name: Steve Lord

Aliases: stevelord (EAB)

Contact info: stevelord@whatiff.info

Websites: www.thedorkweb.substack.com

johnscolieri

Name: John Scolieri

Aliases: klx300r (EAB)
  328gts (amigans.net)

Contact info: johnscolieri@whatiff.info

Websites:

kevinsaunders

Name: Kevin Saunders

Aliases: Invent (EAB)

Contact info: kevinsaunders@whatiff.info

Websites: www.patreon.com/KevinSaunders
www.youtube.com/KevinSaunders
www.twitter.com/invent71
www.instagram.com/invent71/
www.ko-fi.com/kevinsaunders

maddi

Name: Maddi

Aliases: MsMadLemon (YouTube)

Contact info: msmadzie@gmail.com

Websites: www.patreon.com/msmadlemon
  www.youtube.com/c/MsMadLemon
  www.twitter.com/MsMadLemon
  www.facebook.com/msmadlemon
  msmadlemon.bandcamp.com

carlsvensson

Name: Carl Svensson

Aliases:

Contact info:

Websites: www.datagubbe.se

ivansorensen

Name: Ivan Sorensen

Aliases: Weasel Fierce

Contact info: ivansorensen@whatiff.info

Websites: everydayamiga.posthaven.com

danielreimann

Name: Daniel Reimann

Aliases:

Contact info: daniel_reimann@freenet.net

Websites: Translator for www.amiga-news.de