

Apple2000

COLLABORATORS

	<i>TITLE :</i> Apple2000		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		August 24, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Apple2000	1
1.1	Apple 2000	1
1.2	Introduction	2
1.3	Requirements	2
1.4	Description	3
1.5	Why An Apple][Emulator?	4
1.6	Running the Emulation	4
1.7	Loading/Saving Disks/Files	5
1.8	Transferring Apple Files/Disks/ROMs	6
1.9	Paddle/Joystick Emulation	7
1.10	Speed Regulation	9
1.11	The Status Bar	10
1.12	Tech Notes	10
1.13	Planned Improvements	11
1.14	What About EMPLANT?	12
1.15	About the Author	12
1.16	Payment	13
1.17	Credits	14
1.18	History	15
1.19	Finding Software	16
1.20	Compatibility	17

Chapter 1

Apple2000

1.1 Apple 2000

APPLE 2000 v1.3

The premier Apple][emulator for the Amiga
Copyright © 1994, by
Kevin Kralian

All Rights Reserved

Introduction

ABOUT THE AUTHOR

Requirements

CREDITS

Description

HISTORY

Why An Apple][Emulator?

COMPATIBILITY

Running the Emulation

FINDING SOFTWARE

Loading/Saving Disks/Files

Transferring Apple Files/Disks/ROMs

Paddle/Joystick Emulation

Speed Regulation

Status Bar

Tech Notes

Payment

What About EMPLANT?

1.2 Introduction

This program is freely distributable, as long as this instruction ↔
file is
kept with the program, and no modifications are made to my program or
instructions. I grant people the right to use this program privately,
however, it may not be included as part of any commercial package.

This program is FREEWARE (well, more accurately, Tech-Ware). I do not
expect a monetary payment, however, donating me useful, enabling technical
material will result in me creating other emulations...

[click here for more info](#)

NOTICE: Although this program is distributed as Freeware, ↔
copyright laws
& protection still apply. As such, *any* infringement upon this code,
especially as applied towards other 6502 or Apple II emulations will be
rigorously pursued via legal channels.

Standard Disclaimer: This program is AS IS; use it at your own risk! I
assume no responsibility if this program or its use should cause something
disastrous to happen (like your computer exploding or you getting killed.)

I may be contacted at "Kevin_Kralian@sacbbx.com"

This program uses "ReqTools.library", Copyright © by Nico François.

"Disk2File" program Copyright © by Ron Menelli.

1.3 Requirements

REQUIREMENTS:

- o Amiga computer with Kickstart 2.0 or newer
- o A 68020+ CPU. Emulation WILL NOT WORK on a 68000 system at this time.
- o About 900k free RAM (preferably most of it FAST RAM)
- o ReqTools.library by Nico François
- o Apple][ROM image (called _APPLE.ROM)

Recommended:

- o A two-button joystick (to emulate the Apple's two-button joystick)
- o A 68020 at ~25MHz (for full speed 1 MHz emulation)

1.4 Description

DESCRIPTION:

"Apple 2000" is the premier Apple][emulator for the Amiga computer. At its current level it accurately emulates a 64K Apple][+, including:

- o 6502 CPU
- o ALL video modes (Text, LoRes, HiRes, Mixed modes, etc)
- o 16k RAM card (64k computer)
- o 5¼" disk drive (1 or 2, via disk images)
- o Two button joystick / Koala-Pad / Paddles
- o Controllable Speed Regulation
- o Keyboard
- o Sound

The emulation also runs in a completely system friendly manner, multitasking properly with other programs. The two main goals were speed and accuracy. This was accomplished by hand coding the emulator in 100% machine language, optimization via instruction cycle analysis, and painstaking attention to Apple hardware details.

I feel confident that this is the fastest, most complete Apple][emulator available for the Amiga computer (commercial, public domain, or otherwise). Some of the highlights of my emulation:

- o Apple 2000 video emulation is the most accurate around:
 - There is no "dithering" of the 16 Lo-Res colors.
 - The text supports inverse and flashing characters.
 - Two consecutive color pixels are drawn as white (as the Apple does).
 - There are no missing, skipped, or fat vertical lines on Hi-Res gfx.
 - Identical text character set.
- o Disk drive emulation supports loading of "disk images" from any amiga device (no custom archives or conversions required), including:
 - Dalton Disk Disintegrator (DDD) 2.1 format (like DMS on amiga)
 - 143,360 byte "Generic Disk Image", found on the Net & used by other Apple][emulators on other platforms (Unix, Mac, & PC based)
- o Apple 2000 disk drive emulation saves disk images in the standard "Generic Disk Image".
- o Apple 2000 is able to instantly load and run Apple executable files from any Amiga device (better than real Apple; no disk booting reqrd!).

1.5 Why An Apple][Emulator?

WHY AN APPLE][EMULATOR?

Why the Apple][? Sentimental reasons. It's the computer I grew up with and learned to program on. Since I have a fundamental understanding of the Apple and because there aren't any other useable Apple emulators out there (I've seen 5 or 6), the task called to me. I wanted to be able to play all of my favorite games that I grew up with. Yes, they certainly are not cutting edge as far as the graphics and sound goes, but they certainly are playable! And I can overlook the cosmetics for some good gameplay (i.e, just like people appreciate classic cars or oldies music). Plus I wanted all of my friends to be able to play all of those great forgotten games...the classics! The original CASTLE WOLFENSTEIN, CHOPLIFTER, KARATEKA and CARMEN SANDIEGO. How many other multitasking versions of JUNGLE HUNT or ROBOTRON 2084 can you play while downloading a program? By writing this one emulator, the entire Amiga community is suddenly presented with over 10,000 (now multitasking) Apple][programs we wouldn't have otherwise been able to use (or play).

After letting the idea stew in my head for 6 months, and much apparent rambling to my friends (who so nicely encouraged me by saying, "What? YOU write an emulator? And in C? UGH!"), I began coding. One month later, I brought my first creation over to a friends house to see how it worked on his system. After starting it up, we sat there. 30 seconds later we were still sitting there, looking at a white screen. Eventually, we watched as each little white character s-l-o-w-l-y was replaced by a black space. Two minutes later, after getting bored of waiting for it to finish clearing the screen, we gave up and played 2-player LEMMINGS. I knew the only way I was going to be able to make this program 'practical' was to do it in assembly.

I finally bought DevPac 3. After writing a program to bounce 65,535 colored pixels around a screen, I felt ready and experienced. I began converting my routines for my emulator into assembly code. Almost two years later (and after rewriting most of my emulation 10 times) my emulator has finally matured enough to go out into the cold and brutal world. Here it is, ready to be challenged by thousands of Apple programs I have never even heard of, and ready to do its damndest to run them all!

1.6 Running the Emulation

RUNNING THE EMULATION

Make sure "ReqTools.library" is in your libs: directory and place "Apple2000" and "_APPLE.ROM" in the same directory. Then from the CLI/Shell, CD to its directory and type "Apple2000" (or use it's icon). If you want to run the emulation with two emulated disk drives, type "Apple2000 -2" (or modify the startup script used by the icon).

Now, assuming a little common sense (press the "OK" button on the window!), you will see a black screen with words "Apple][(" at the top. Congratulations, you are now using an Apple][computer. The Apple is trying to boot a disk.

I will assume you have a little knowledge on using an Apple][. Here are some of the pertinent keys:

KEY	Function
DEL	Apple "Reset" key.
ctrl-DEL	Similar to "Ctrl-Open Apple-Reset" on][e,][c,][gs. Forces reboot, even if reset vectors have been changed.
RAmiga-Q	Quit the emulator (after verification).
RAmiga-L	Load Apple disk image or executable into the emulator.
RAmiga-S	Save Apple disk image.
L-ALT	(Like Open-Apple on][e) Represents Apple Paddle Button #0
R-ALT	(Like Closed-Apple on][e) Represents Apple Paddle Button #1
F9	(Alt keys do not affect other keystrokes to emulator) Toggle between Mouse / Joystick control.
HELP	Sequence through simple help-messages.

Speed Regulation:

F1	50% Apple Speed (0.5 Mhz)
F2	100% Apple Speed (1.0 Mhz)
F3	150% Apple Speed (1.5 Mhz)
F4	200% Apple Speed (2.0 Mhz)
F5	Unlimited (as fast as your system can go!)
^	
	Arrow keys patched to be like Apple][e,][c,][gs.
<--+-->	(Note: Apple][+ had no Up/Down arrows, and most older programs won't handle them as expected.)
v	

NumPad only:

8	Trim Apple joystick center position in respective directions.
4 5 6	
2	"5" will reset it to default (of 127,127).

1.7 Loading/Saving Disks/Files

LOADING DISKS/FILES

Once the Apple is running, you'll probably want to load an Apple disk or executable. Here's how: at ANY time during emulation, feel free to press Right-Amiga-L to bring up the Load File requester. From this requester you may load Apple 5¼" disk images or executable files. Simply navigate to wherever the files are kept and load the file/disk image you want. Apple 2000 recognizes several types of load files:

- o Filenames with a <xxx> suffix are Dalton Disk Disintegrator archives (DDD was a common disk compression util for the Apple, similar to DMS for the Amiga) and the emulator will automatically decompress them!
- o 143,360 byte files are assumed to be "Generic Disk Images", such as the many available on the net.

- o Files with the proper internal Dos/ProDos header are executable files; these are single files that were runnable from Apple DOS 3.3/ProDos and did not require any disk access thereafter. These files now do not even require booting any Apple disk and are simply loaded into the appropriate Apple memory areas and started instantly (quicker and easier than a real Apple!).

If you started Apple 2000 with the 2 disk drive option (via -2 option), you will be asked which drive you want to load the image into. After loading a disk image into drive #1, the emulator will ask you if you want to 'boot' the disk. If you choose not to, you have effectively just 'put the disk in the drive' (useful when you need to insert 'Disk 2'). On the other hand, loading an executable Apple file does not give you any choices and immediately runs it. This has all been designed to keep the emulator as clean and simple as possible in terms of starting and running Apple programs for the non Apple-literate user.

Keep in mind, loading a disk image is the same thing as inserting the disk into the Apple drive. It will STAY there until you replace it with another disk (or some program erases that disk). Even after you load and run several executable Apple programs, hitting Ctrl-DEL (rebooting the Apple) will boot up the last DISK IMAGE you loaded (if any). This can be confusing if you don't know what's going on (i.e, after finishing playing MS. PACMAN and resetting the Apple, why is MUSIC CONSTRUCTION SET loading? Because the disk is still in the drive from before).

SAVING DISK IMAGES

Pressing Right-Amiga-S will bring up a requester to save a disk image. Disk images are saved in either the 143,360 byte "Generic" format, or in a semi-compressed DDD format! The determination is made by the filename. To compress a file in the DDD format, the filename must end with a '>' char. Otherwise, it will be saved in the 143,360 byte format. If you do save the image that way, it is suggested you append a ".disk" onto the filename to maintain naming conventions (and so the requester will show the file).

Keep in mind, DDD compression is not as tight as external compression programs (LHA, etc). Also, loading/saving DDD files takes a couple extra seconds to perform the (de)compression.

1.8 Transferring Apple Files/Disks/ROMs

TRANSFERRING APPLE FILES

To get an executable binary file from a real Apple to the Amiga is quite simple. Use a terminal program and then transfer it via null-modem (or however you want) to the Amiga. It's recommended you attach a .PROG to the end of the filename for consistent naming conventions.

TRANSFERRING APPLE DISKS

Only UNPROTECTED standard 16 sector Apple disks are currently useable.

This eliminates copy protected software. Simply run Dalton Disk Disintegrator (do not use version 2.0! It has a bug! Use version 2.1) and use it to compress the disk into a file. Then, transfer it to an Amiga (as described in "Transferring Apple Files" above). Once on the Amiga, ensure the filename ends in <xxx> so the emulator recognizes it as a DDD file.

TRANSFERRING APPLE ROMS

The Apple emulator, being true to form, requires the actual Apple ROM data in order for the Apple to do anything. The standard Apple ROMs in use were the 'AppleSoft ROMs that contained AppleSoft BASIC, the assembly language monitor, and autobooting code. So I suggest that you obtain the same ROM if you would like the same compatibility. The ROM image can be obtained by booting an Apple][or][+ with DOS 3.3, then typing:

```
BSAVE BASICROM,A$D000,L$2FFF
```

to save it to disk. Also, the disk controller Rom (not required) can be saved by typing:

```
BSAVE DISKROM,A$C600,L$00FF
```

Incidentally, the main ROM image is on Apple's "DOS 3.3 System Master" disk, called FPBASIC (which may be used instead). After saving these images to disk, use your favorite terminal software and a null modem cable (or real modems or whatever you like) to transfer these files to the Amiga. Once transferred to the Amiga, give these files the proper names and place them in the same directory as the Apple2000 executable.

Theoretically, you can use the ROMs obtained from an Apple][clone (i.e., Franklin Ace, PineApple, etc.), but keep in mind that these ROMs were not 100% compatible (but were quite close). This would effectively make my emulation a "Franklin Ace Emulator". :-) However, you CANNOT use the ROM images from an Apple][e,][c, or][gs (maybe eventually...we'll see)!

Once on the Amiga, the main Rom MUST be named "_APPLE.ROM" and the disk rom (if you have it) must be called _DISK.ROM.

** NOTE **

Starting with version 1.1, due to the difficulties people had trying to obtain it, the _DISK.ROM is no longer required! If that file is not present, Apple2000 will automatically use a custom-disk rom instead. However, obtaining the true rom image is still recommended for 100% compatibility with all software.

These ROM files can also be found in several other Apple emulator archives available on the net and on several CD-Rom compilations (including the PC Apple][emulator called "Apple2Em.zip").

1.9 Paddle/Joystick Emulation

PADDLE/JOYSTICK EMULATION

The Apple][commonly uses either two paddles, an analog joystick, or a graphics tablet (like a free-floating joystick). My emulation covers all bases. Apple 2000 supports using these control devices:

- o Amiga Joystick (preferably 2 button, to emulate apple joystick)
- o Amiga Mouse (to emulate free-floating joystick / paddles / Koala-pad)
- o Analog PC Joystick (via FighterDuel dongle, to emulate apple joystick)
- o Atari Paddles (to emulate apple paddles)

The F9 key toggles through the choices and displays your selection at the bottom of the screen.

Amiga Joystick:

The emulator can utilize the standard amiga/atari/sega joystick (preferably, a true 2-button joystick), and will assign the 8 switch positions (up, up-right, right, etc) to the extreme values that an analog stick would return (0 & 255). The "center" position defaults to the optimum values (127 x 127), however, some games expect different center values. This center value can be trimmed with the 2, 4, 6, & 8 keys on the -numeric keypad- ONLY (see "Running Emulation"). For example, if you start CHOPLIFTER and your character drifts towards the left, press the "6" key to center the joystick more towards the right until your character no longer drifts! (This is the emulated 'joystick trim' function).

If you do not have a two button joystick, you have two choices. Spend \$8 and buy one, or else just use the Right-ALT key in lieu of the second button (by the way, the two ALT keys work great for pinball games like RASTER BLASTER). Note that the Sega Genesis Game controllers work well with the amiga, and have two buttons!

Amiga Mouse:

The emulator can utilize the absolute mouse coordinates, and this works well for simulating a Koala-Pad graphics tablet, free-floating joystick, or paddles (left/right control paddle #0, up/down control paddle #1).

Programs designed for graphics tablets (KOALA PAD) or un-centered joysticks work great in this mode, i.e, FANTAVISION, MISSILE COMMAND, MUSIC CONSTRUCTION SET, and most other free floating cursor control programs work ideally. The mouse works just like you would expect here.

Analog PC Joystick:

The emulation also supports true analog joysticks via joystick-dongles (such as the adapter used for Fighter-Duel). The joystick will work just like a real analog stick.

*** Note: Apple 2000 uses "dynamic range setting & calibration" for analog joysticks. This means that when you choose the "Analog Joystick" option, the first thing you must do when starting to play your game is to move the joystick over its full range of motion (extreme left, right, up, & down). Once that is done, you need not worry about it anymore (unless you turn off "Analog Joystick" and select it again). Joystick trimming is to be

done with the joystick's own trim pots. This 1 second of work from the user ensures that Apple 2000 will work with a wide variety of joysticks.

Atari Paddles:

 The emulator also now supports using actual Atari Paddles (the kind used in the old Atari VCS/2600). They plug right into the amiga game port, and work just like real apple paddles. NOTE: Like the analog joystick, the paddles need to be moved over their complete range of motion when you start to use them.

Other game controller notes:

 Some of the older Apple games were designed to be used with paddles, not joysticks. This is noticable in games (APPLE GALAXIAN, SNEAKERS) as in when you release the joystick, your ship automatically moves back towards the center point on the screen. This is exactly what would happen on a real Apple with a joystick. You need to use some type of paddle emulation here, either via the "Amiga Mouse" or "Atari Paddles" options.

*** WARNING! While it is a bad habit many people have of (un)plugging game controllers while the computer is on, be aware of this potential disaster: Some of the Analog-Joystick Adapters have a metal jacket around the front plug. This metal sheath has the potential to short out pins on the amiga game port while trying to plug it in, which can damage your amiga! The solution is simple: turn off your computer while changing controllers. "An ounce of prevention..."

1.10 Speed Regulation

SPEED REGULATION

"IT'S TOO FAST!" :-)

By popular request, Apple 2000 now offers selectable speed regulation so people with fast systems can slow down games to humanly playable speeds. Speed regulation is controlled via 5 function keys (F1 -> F5), which are in order from slowest to fastest.

F1	50% Apple Speed (0.5 Mhz)
F2	100% Apple Speed (1.0 Mhz)
F3	150% Apple Speed (1.5 Mhz)
F4	200% Apple Speed (2.0 Mhz)
F5	Unlimited (as fast as your system can go!)

F2 selects "100% Apple Speed", which limits emulation speed to that of a real Apple][(1.024 Mhz). F1 selects 50% (slow down games to half speed), and F5 selects "unlimited speed" (which runs as fast as your system can).

Use of these keys during gameplay can be used as an aid- For example, switching to 50% speed during a difficult part in an arcade game. Also, some games on the apple might be a little sluggish compared to today's standards and skills. 150% speed usually takes care of those. :)

Understand that Speed Regulation works like a "Speed Limit". If your system can't run the emulation quickly in the first place, choosing 150% or 200% regulation won't have any noticeable effects (Like taking a '74 Pinto out on the Autobahn. Who cares if there's no speed limit- You won't go very fast. :)

Keep in mind that Apple sound is made via speaker clicks and timing loops. Faster/slower emulation will cause higher/lower pitched sound.

"IT'S TOO SLOW!" :-(

Okay. For the remaining 93% of the Amiga owners who don't have '040s, there are a few things that can be done to speed up the emulation.

Press F5 to turn off speed regulation- This completely turns off the overhead of the speed-regulation code.

If you have an MMU (and an additional 512K free fast-ram), turn on the "FastRom" option. (Type "CPU FASTROM" from the shell).

If you are running ENFORCER, turn it off! On my system, graphics intensive games slow down 20% while enforcer is running.

Start "Apple 2000" in a relatively clean system with large blocks of fast-ram available. (IE: Instead of trying to run the emulator while you've been working on your system for 8 hours and have DirOpus, AMAX IV, and ADPro all running, just reboot the system first).

Of course, you can always add an accellerator to your system. ;-)

1.11 The Status Bar

THE STATUS BAR

By popular request, the "Title Bar" at the top of the screen has been removed, and has now been replaced with a new "Status Bar". The bar appears at the bottom of the screen momentarily to notify the user of any significant events, and then goes away. And it's black and white so the grey scale users can read it.

This bar displays significant messages, including feedback on Speed Regulation Selections, Paddle/Joystick Controls, Apple Crashes, Help, and disk drive access (including drive & track #'s). Most the time the status bar will be invisible.

1.12 Tech Notes

TECH NOTES

Some Apple programs use "unimplemented" 6502 instructions. These are

instructions that are not official, but partially decode into doing a particular function (as discovered by many unorthodox programmers). My emulation does not support ANY unimplemented instructions, and will simply break upon hitting any of those instructions (with Apple][software, I have seen very few programs that use them). Those few programs will also fail on the Apple][gs.

The ONLY graphics glitch is that the Hi-Res graphics screens do not fill in the entire display width. That is, they leave a half-inch black border on each side of the display. Why? Because the Apple Hi-Res screen has a horizontal resolution of 280 pixels, and the Amiga's display has a minimum resolution of 320. Trying to stretch this display by leaving an empty pixel after every 7 pixels or drawing every 7th pixel twice, results in a highly distorted and uneven image. The Text modes and LoRes modes still use the entire screen width (to maintain aspect ratio). This slightly narrow display is only noticeable in the 'mixed Graphics/Text' mode, where text will be a little wider than the graphics above it.

Also regarding Text and Graphics (but not a glitch, it's an improvement) is the fact that mixed Graphics and Text on the old Apple]['s originally caused the text to be fringed with green and purple instead of being solid white. This fringing has absolutely no purpose, but is a mere artifact of the Apple video circuitry. My emulation cleans it up with crisp & clean text output at all times (does anybody have any complaints?). Apple finally cleaned this up with the Apple][gs and its RGB output (but introduced a couple other graphic glitches), so I believe my clean Text display is desirable.

1.13 Planned Improvements

PLANNED IMPROVEMENTS

Currently, emulation speed is pretty much as fast as possible under the current 'interpreted' method. A speedup of about 3X is possible if I do 'pre-interpretation' which essentially converts 6502 code to native 680x0 code ahead of time, then running it at full speed. However, this comes at a cost of excessive memory usage (I estimate using about 2 megs for the 64k Apple emulation). This could be considered if enough people are interested (this would be the final speed boost required for the A1200 owners if they have enough memory!), but is of low priority for now.

A 68000 version is easily possible, but emulation is so slow at that point (games are frustratingly unplayable), that I haven't bothered to do one.

I plan to add Apple printer & serial emulation (and redirection). This way you could redirect Apple printer output to an Amiga file or to an Epson emulator (to print Epson output to any Amiga Prefs printer). Or emulate the Apple serial card with an Amiga Modem, etc.

I plan to (eventually) upgrade the entire emulation to Apple][e /][c status. This includes Apple "Double-Hi-Res" graphics, 128k RAM, and 80-column text.

I have thought about writing a ProDOS driver allowing the Apple to access Amiga devices as an Apple hard drive (is anybody using the emulator this

seriously?).

I might tackle using a real 5¼" disk drive if enough people want it.

1.14 What About EMPLANT?

WHAT ABOUT EMPLANT?

My emulator, "Apple 2000" was (p)reviewed in Amiga Computing (Issue 71, March 94), inside a larger review for the Emplant card (there's even a screenshot where you can read my title bar, Apple 2000!). For all intents and purposes, the review makes it APPEAR as if this program was written by, owned by, and coming soon from Utilities Unlimited, makers of the Emplant card (A macintosh emulator). Regardless, the reviewer loved it, noting that this was the fastest 6502 emulation he has seen.

At several World Of Commodore shows, Jim Drew showed my early versions of "Apple 2000" to crowds of people during his presentations of his Mac emulator. A friend even has a video-tape of Jim loading up and showing my emulator to a crowd when I asked, "What other emulators are you doing?" (before he knew who I was) at WOC in Pasadena, 1993.

To set the record straight, I did send Utilities Unlimited several early exclusive 'evaluation' versions of my emulator to see if they were interested in purchasing it (for their Emplant package), but no agreements were ever reached. Utilities Unlimited was in no way involved with the development of, and has no connection to, this program (Apple 2000).

This program is NOT part of the Emplant package, as many people have been led to believe. The positive side of this is that you may use Apple 2000 without having to spend >\$300!

1.15 About the Author

ABOUT THE AUTHOR

"Apple 2000" was written by Kevin Kralian over the course of two years. After serving in the US Marine Corps Infantry (!), he then earned a college degree in Fire Technology, and continued in school towards his Computer Science degree. He is also a volunteer FireFighter/EMT and a freelance programmer.

He has over 10 years of programming experience, including ADA, BASIC, C, Pascal, 6502 and 680x0 assembly. Programming interests focus on performance programming, including games and emulation. Career goals include firefighter and game programmer.

He has recently been hired as a full time game programmer, and is now working on the Sega "Saturn". Good things do happen to good people. :-)

He may be contacted at "Kevin_Kralian@sacbbx.com"

1.16 Payment

"PAYMENT" FOR THIS PROGRAM

This program is being distributed as Freeware, I do not expect monetary payment. My original intentions were simply to have my program be 'used' by the Amiga community, and I still feel the same way. I've worked long and hard on this program and the most rewarding thing to me know would be to simply know people are enjoying it!

However, what I WOULD appreciate would be any technical references for any computer/hardware/platform. Let me explain...

Many improvements in the Apple emulator are dependant upon me finding Apple technical reference material (i.e, unimplemented instructions, serial/parallel support, ProDOS harddrive support, etc). If you would like to see these features implemented, the biggest thing you can do is send me any tech material that could be helpful.

Also, some ideas for my next emulator include: Atari VCS (2600), GameBoy, Nintendo, Atari 400/800 and Commodore 64/128. Though there are a few C-64 emulators out there, many people have urged me to do one "the right way". I tend to want to do the old Atari VCS or Gameboy emulation. HOWEVER, in order to do this, I need tech information that I cannot publically obtain.

Do YOU want these game machines to be emulated (I do)? If you are one of those priviledged people who might have been involved in developing software for any of these machines or somehow have any tech info on these machines, please send me any and all tech information. *** I WILL *** make an emulator of these machines when I have enough tech information to do so. But I need your help.

I am open to any suggestions, comments, or feedback. Let me know how the emulator works for you. Please let me know of anything that does not work (that works on a real Apple][), and I will do my best to correct the problem. I am also interested in obtaining any Apple][programs people may have to test under my emulation.

Anybody interested please contact me at "Kevin_Kralian@sacbbx.com"

Particular things I'm looking for:

- o Whats Where in the Apple][: An Atlas to the Apple computer
- o Apple Super Serial Card / Parallel card manuals
- o AmigaDOS Programmers Reference
- o Any 2.0+ Amiga AutoDocs (AmigaGuide format would be cool!)
- o ANY kind of tech info on Gameboy, Nintendo, or the old Atari VCS (there once was an Apple][card to program the Atari. Anybody have it?)
- o Any and all Apple][programs.
- o Any responses, reactions, suggestions, etc. on my emulation
- o etc...

1.17 Credits

CREDITS

I owe lots of thanks to lots of people.

Thank you my dearest JoAnnaBear for being so supportive of me and this project over the last two years, and for not going crazy over my many hours of "techno-babble", but just patiently smiling back as if you understood me. :-)

Thank you Robbie for all your inspiration and encouragement. And thanks for your brainstorm sessions and hundreds of hours worth of second-hand smoke (cough cough). Thank you for the book "Amiga Machine Language Programming Guide" - the very first 680x0 assembly book I've seen (blech!). By the way, this book was due back at the library in 1989! How are your games "To Sir With Love" and "The Piano" coming along? Oh yeah... and thanks for cleaning up and converting my docs to AmigaGuide format for me :-)

Thanks to Ronald J. Menelli, for his contribution of the "Disk2File" conversion utility. ("Disk2File" is Copyright © by Ronald J. Menelli).

Thank you Brian J. Bernstein, for his beautiful new Apple 2000 icon. And thanks to the many others who sent icons. Apparently people did not consider the little face representative of Apple2000? ;-)

Thank you so much to Rand, another Firefighter, for recognizing a good thing when he sees it. (Whats with all these firefighter/programmers?)

Of course, Thanks to Jason Compton (Amiga Report columnist), the man who did the first [authorized] review of Apple2000, and informed the public of this great program. How could I have forgotten you for so long?

Thanks Ed Brown, your support and encouragement has been exceptional! You better be happy with the new "Analog Joystick Support" after telling me about the advantages over the "digital" sticks! ;-)

Thanks to "Nine Inch Nails" and "Frankie Goes to Hollywood" for their soothing sounds to which much of Apple2000 was created.

Thanks to the authors of other Apple][emulators on other platforms, for their open discussions, comparisons, and sharing of emulation techniques and technology. Their willingness to disclose results in improved software for all computer users.

Thank you to the many people who have contacted me about the initial release of "Apple 2000". All of your letters of support, feedback, contributions, suggestions, etc. are enthusiastically welcomed!

Thanks to those incredible guys at Computer Cafe. I appreciate how you let me use your various machines for debugging and testing during the development of my emulation. Without your help, I would have never been able to work out the '040 bugs, nor have seen my emulation running on a 28" monitor with cool 24 bit backgrounds.

Thanks to Carmen Rizzolo, the computer artist extraordinaire! Your original artwork for my previous programs are utterly amazing. Without people like Carmen, where would we get cool 3D Star Trek and telephone objects?

Thanks to Will, the only intelligent Mac owner I know. It was great to share ideas on high performance 6502 emulation with the 680x0. Have you finished your Mac version of your Apple][emulator yet? Thanks for that 'half' of the "Inside the Apple //e" manual. Did you ever find pages 1-110?

Thank you Nico François, for your contributions to the Amiga community. ReqTools is a very polished piece of work, and I know that your work has saved me (and many others) hours of work trying to "recreate the wheel". (Reqtools.library is Copyright © by Nico François).

Thanks to the many helpful people on the Internet, for helping me through many obscure programming and debugging challenges.

Thanks to Steve Wozniak for creating the original Apple][. And congrats to Apple Computer for knowing how to market computers and becoming a large, successful company. Maybe Commodore can learn a few things from you before they drive themselves out of business? [May '94- too late]

Thanks to 'Dalton', for his "Dalton's Disk Disintegrator" (DDD) program on the Apple][. My (de)compression routines were based on his routines and attempt to compress data in an identical, compatible way.

Thank you Bill, for taking your family and moving far, far away.

Thanks to an unnamed individual, for teaching me a very important lesson- That I can never trust anybody, and that people like you will lie, cheat, and steal to try and build the credibility of yourself/your company. Just another page in your book of lies, eh?

And finally, thanks to the many people I do not have space to mention, and to all of the Amiga users who have made the Amiga scene as wonderful as it is.

1.18 History

HISTORY

08/21/94 ***** Version 1.3 *****

- o Controllable Speed Regulation (50%, 100%, 150%, 200%, Unlimited)
 - o Overhauled entire video subsystem
 - No more hardware hitting or copperlists
 - Works happily with AGA/Mode promotions
 - No more "Screen Jitter" for the few who had it
 - Screens can be pulled down in front without disturbances
 - All Apple screens can now be "Grabbed"
 - o Added optional 2 disk drive support (via -2 command)
-

- o Overhauled File Loading/Saving related functions
 - With 2 drives, loading/saving will prompt for drive #
 - Saving: The proper "loaded" disk name will appear (for each drive)
 - Saving: DDD vs 143,360 byte formats chosen via filename.
 - Added "Warning: Disk Data Changed" notice during loads & quit.
- o Replaced TitleBar at top of screen with "Status Bar" at bottom
 - Only shown during significant events
 - Black and white (so greyscale users can read it)
- o Added support for Analog Joystick w/ 2 buttons
- o Added support for Atari Paddles
- o Video page-flipping handled properly
 - ShortCircuit, BileStoad, & OutPost run much better!
- o A couple bug fixes (and duplicated a 6502 bug thats required)
 - Canyon Climber, Drol animation, Frogger, Lode Runner, MoonPatrol, Pinball Construction Set, Randamn, Sargon, & Tetris II All work!
- o Added Lower Case Text display (Apple][e charset)
- o Optimized disk loading/decompression routines (twice as fast)
- o Added a cheesy "Turn Off Monitor" effect when exited ;-)
- o And of course, its a little bit faster!

(Version 1.2 skipped due to misnamed archives already out)

05/18/94 ***** Version 1.1 *****

- o Wrote custom disk Rom that is used if _DISK.ROM file is not present! (_DISK.ROM is no longer required! But still recommended...)
- o Check for 68020+ CPU (instead of crashing)
- o Support for 143,360 byte generic "disk images" (Loading/Saving) (as used by many other emulators & available on the net)
- o Disk images saved in "generic" format instead of DDD (DDD format still recognized & loaded, though)
- o Dos 3.3/ProDos headers recognized and handled in both disk images and executable files (instead of just ProDos like version 1.0)
- o Dos 3.3/ProDos/Raw ROM images all recognized and acceptable.
- o Inclusion of "Disk2File" utility, which reads a 5¼" Apple disk via the C= 1020 Drive, and saves it as a file
- o Included "Version" string
- o Numerous optimizations (-slightly- faster)

04/10/94 ***** FIRST PUBLIC RELEASE v1.0 *****

1.19 Finding Software

FINDING YOURSELF SOME SOFTWARE

Finding apple disk images is like a treasure hunt. If you can't transfer files yourself, there are lots of apple files out on the Net and BBS's.

I'm told you can find lot's of disk images on anon ftp:

wilbur.stanford.edu:/pub/apple2/disk_images and /pub/apple2/incoming
ftp.uni-kl.de:/pub/apple2/disk_images
cassandra.ucr.edu:/pub/apple2/incoming

FSP site, at: 134.184.15.12 2424
At (unknown) site at: minnie.sc.adfa.oz.au

Expect difficulties! Some of the images have Mac headers on them that need to be removed (Leave it to the mac to screw up a simple data file). Some are compressed with Mac, Apple, or Unix protocols. Some have out-of-order sectoring and need to be re-mapped (via AFID utility). Some files are "text" files that try to install themselves via "executing" from the apple. And many archives are simply corrupt and won't even work on real Apples!

The best way to avoid the headache of sifting through so much garbage is to trade with other people who have known, tested, and working programs. I've even been informed of several huge archives or commercial games for Apple 2000 floating around many BBS's.

That brings out another issue. Most these games are/were copyrighted, commercial programs. But now what? Most the companies have since dissolved, none of the apple programs are sold or produced anymore, and who retains the copyright is hard to determine. This very issue has been controversial for quite some time, and there is no easy answer.

What is known is that many of the classic apple programs have been placed in public domain by their authors (and many agree to it just by asking). Some other companies have even placed all their old Apple titles into public domain (as written in the late issues of A+ or InCider). The sample programs I include with my archive, to the best of my knowledge, are now in the public domain (according to statements in magazines or what's sold by PD distributors).

1.20 Compatibility

COMPATIBILITY

Every effort was made to provide the highest degree of compatibility with all Apple][software. This is a small sampling of the more than 600 programs that I have successfully ran under Apple2000.

AIR_CARS, ALIEN_AMBUSH, ALIEN_TYPHOON, ALPHAPLOT, APPLE_PANIC, APPLE_ZAP, APPLECIDER, APPLEIIIeINTRO, ASTEROID_FIELD, AUTOBAHN, BAG_OF_TRICKS, BEAGLE_BAG, BEER_RUN, BEYOND_CASTLE_WOLFENSTEIN, BIG_MAC_ASM, BLISTERBALL, BLITZKREIG, BOLO, BOULDER_DASH, BUG_ATTACK, BUG_BATTLE, BUZZARD_BAIT, CANNONBALL.BLITZ, CENTIPEDE, CHAMPIONSHIP_LODE_RUNNER, CHIVALRY, CHOPLIFTER, COMPUTER_FOOSBALL, CONAN, COPY_II+_7.4, COUNTY_CARNIVAL, CROSSFIRE, CUBIT, CYCLOD, D-CODE, DEFENDER, DIG_DUG, DINO_EGGS, DOGFIGHT, DOS_3.3_SYSTEM_MASTER, DRAGONFIRE, DRAW_POKER, DREADNOUGHTS, DUNG_BEETLES, ELIMINATOR, F-15_STRIKE_EAGLE, FALCONS, FANTAVISION, FIGHT_NIGHT, FIREBIRD, FISHIES, FLIGHT_SIM_II, FRAMEUP, FRAZZLE, FRENZY, GALAXIAN, GOLD.RUSH, GRAPHICS_MAGICIAN, GREAT_CROSS_COUNRY_ROAD_RACE, GUARDIAN, GULF_STRIKE, HACKER, HACKER_II, HARD_HAT_MACK, HARDBALL, HUNGRY_BOY, INTERLUDE_II, INTERNATIONAL_GRAN_PRIX, JAWBREAKER, JPORT, JUMPJET,

JUNGLE_HUNT, KARATEKA, LABYRINTH, LEATHER_GODDESS_OF_PHOBOS, LEMMINGS,
MARAUDER, MARIO_BROS, MASQUERADE, MICRO_ILLUSTRATOR, MILLIONWAIRE,
MINE_SWEEP, MINER_2049ER, MR.DO, MS_PACMAN, MURDER_ON_THE_ZINDERNEUF,
MUSIC_MAKER, MUSIC_CONSTRUCTION_SET, NEPTUNE, NIGHT_MISSION_PINBALL,
NIGHTMARE.GALLERY, NORAD, PACMAN, PINBALL_CONSTRUCTION_SET, POOYAN,
POPPLES_XMAS_ADVENTURE, PRODOS_MASTER, RASTER_BLASTER, RED.ALERT, REPTON,
RESCUE_RAIDERS, ROADWAR_2000, ROBOTRON, ROCKET.COMMAND, SAMMY_LIGHTFOOT,
SEA_DRAGON, SERPENTINE, SHORT_CIRCUIT, SKYFOX, SNEAKERS, SPACE_RAIDERS,
SPACE_QUARKS, SPACE_WARRIOR, SPY.HUNTER, SPYS_DEMISE, SQUADRON_617,
STAR.THIEF, STAR_WARS_II, STARGATE, STARMAZE, STICKYBEAR_MATH#1,
STICKYBEAR_MATH#2, SUCCESSION, SUICIDE, SUMMER_GAMES, SUMMER_GAMES_II,
SUPER.PUCKMAN, SUPERMAP, SWASHBUCKLER, TAKE1, TAXMAN, TERRAPIN_LOGO,
THIEF, TIME_IS_MONEY, TRANQUILITY_BASE, TRIVIA_FEVER, TWERPS,
UBOAT_COMMAND, XEVIOUS, ZAXXON
