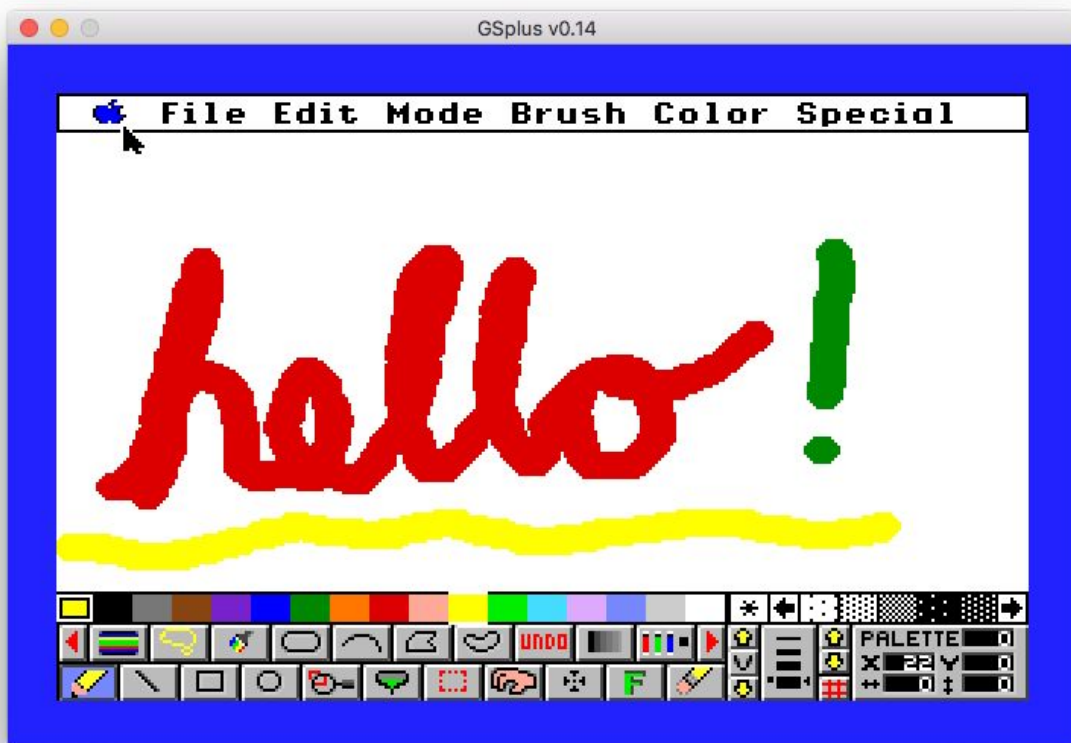




GS+ Reference Manual

A cross-platform Apple II GS emulator



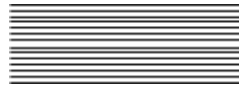
GSplus v0.14
August 14, 2018



Contents



Chapter 1	Overview	3
	A word about the Apple IIGS	3
	The idea behind GSplus	4
	Kegs, GSPort, and GSplus	4
	Currently supported platforms	5
	Future goals	6
	Support and Development	6
	Getting Support	6
	Developer Contribution	6
Chapter 2	Getting Started	7
	Installing GSplus	7
	Launching GSplus	7
	Emulator configuration	8
	Configuration files	9
	Default Configuration Files	10
	Multiple Custom Configuration Files	10
	Configuration menu	11
	Command-line options	12
Chapter 3	Input Devices	13
	Mouse	13
	Joystick	13
	Joystick Options	14
	Keyboard	14
	Apple IIGS keyboard map	14
	Emulator Hotkeys	15
Chapter 4	Advanced Usage	16
	Debugging	16
	Special configuration tips	16
	Troubleshooting	16
Appendix A	Full list of configuration options	18
Appendix B	Apple IIGS Documentation	19
Appendix C	Credits and Thanks	20



Overview



There are a variety of ways to configure your emulated IIGS with GSplus. If you want to get started quickly, skip straight to Chapter 2 - Getting Started.

A few words about the Apple IIGS

“The Cortland is a high-powered addition to the Apple II family. Like Janus, the god of doorways, the Cortland looks in two directions. First, it looks toward the future: With its many high-performance features, such as improved color display, advanced sound system, 16-bit microprocessor, and larger memory, the Cortland makes it possible for future application programs to be more powerful. Second, the Cortland looks toward the past: Because it has the features of the Apple IIe and the Apple IIc, it can run most of the programs written for those computers.”

Allen Watson - Technical Introduction to the Cortland - 1986

The Apple IIGS, codenamed Cortland, was the final generation of the Apple II computer. It was a fully compatible 8-bit Apple II, as well as a modernized 16-bit system that showed a glimpse of what could have been a long-term evolutionary path for the platform. However, the system received few hardware revisions and had a hit and miss design. It suffered from limitations like slow 1MHz graphics RAM, but simultaneously offered advantages like a palette of 4096 colors and a powerful sound chip. These “Graphics” and “Sound” capabilities were the “GS” in Apple IIGS.

The system included some interesting bits of hardware. The Ensoniq DOC 5503 microprocessor can play back up to 32 channels of sounds from its own 64KB RAM area. Another special chip is the Mega II which contains all the discrete logic of a standard 8-bit Apple II. The FPI (CYA) controls synchronization between the “fast” and “slow” sides of the RAM bus. There’s the VGC which generates all the graphics signals for the RGB monitor. And it was the first computer with the Apple Desktop Bus (ADB) microcontroller for keyboard and mouse communication, which later was used in Macintosh for decades until USB eventually replaced ADB.

All of these things make the Apple IIGS an interesting and fun computer but one that is still not accurately emulated. It’s important that someday it is accurately emulated as today’s hardware will not survive that much longer, and already much of the disk, drive, and monitor inventory is failing.

Hopefully, you enjoy the efforts put forth here, supported by many programmers over many years, though they may not be 100% accurate to the real Apple IIGS hardware. We love this platform and this computer, and the desire is that you get to have the same enjoyment of the system that we do.

The idea behind GSplus

Various Apple IIGS emulators have existed for years, but they typically targeted a single host OS, and often suffered from lack maintenance. All had accuracy issues, some more than others.

- ❖ “As an avid Apple IIGS programmer, I needed a better-supported emulator. I needed an emulator with some modern features. The world needed an emulator that would run on Windows, OSX, and Linux without needing a Ph.D. to compile it.” -Dagen Brock

The idea was that it would be great for someone with experience and knowledge in emulator development to step forward and make a new emulator, or fix an existing one. Unfortunately, no one did. Thus, Dagen Brock decided to start this endeavor with GSplus, despite having no discernible skills that would have ever made him suitable to do so.

Kegs, GSPort, GSplus

GSplus is not an original emulator. It is a fork of GSPort, which itself is a descendant of KEGS.

KEGS was an early Apple IIGS emulator by Kent Dickey written in the 1990's. It really is an exceptionally good emulator, especially considering the time when it was written and thankfully it was released as Open Source. This allowed it to eventually be ported to other operating systems like OS/2, Windows (Win32 API), Mac (Carbon), and of course Linux.

GSPort was a very nice project that seemed to add a lot of functionality for things like networking, printing and more configuration options. Along the way, there were additions for another project called ActiveGS, which was a browser-based version.

GSplus was forked from GSPort to try to encapsulate all of the work they had done over the years. There was one main goal that GSplus was intended to achieve:

- ❖ Support Windows, OSX, and Linux with consistent features across all platforms.

To get to that point, many hours of work have been put in over the past few years to:

- ❖ Automate GSplus builds and packages for all OSes
- ❖ Deprecate all the ancient code (sorry OS/2 users!)
- ❖ Move to a single set of code drivers for mouse/keyboard/display/joysticks across all Operating Systems using the SDL2 media abstraction layer.

And of course, there have been countless fixes, cleanups, and small feature additions* along the way.

* And new bugs!

Currently supported platforms

To achieve consistent platform support, the decision was made to focus on one main line of code for the graphics, sound, and input by switching to the multiplatform SDL2 library for all systems that could support it, and adding features to the SDL2 driver set. Code for other drivers, like X-Windows and /dev/audio, still exists. These days, Linux is the only system to install X-Windows by default so X is no longer supported. You can try to build other versions on your own and are encouraged to contribute or fork the project to suit your needs (see Support and Development).

Table 1-1
Officially Supported Builds/Code

Operating System	Driver	Description
Windows	SDL2	Builds available
Mac OSX	SDL2	Builds available
Linux	SDL2	Builds available

Table 1-2
Known working Builds/Code

Operating System	Driver	Description
Windows	Win32	Code available, builds still run
Linux	SDL	Code available, builds still run
Linux	X-Windows	Code available, builds still run

Table 1-3
Unknown driver status

Operating System	Driver	Description
Raspberry Pi	Framebuffer/X?	Believed to be working
HP/UX	X-Windows	Unknown
Solaris	X-Windows	Unknown
Mac	Carbon	Deprecated / Unknown
OS/2	Framebuffer?	Deprecated / Actively removing!

Future Goals

Getting GSplus running across all major desktop operating systems was a primary goal, with the idea that it would make things easier to develop a single codebase. A secondary goal has been an enhanced debugger. There is support for this debugger in current builds, but it is not documented at this time. This debugger allows for remote debugging using a special telnet protocol. The work on this is intended to resume shortly.

Fixing accuracy errors and making certain changes is difficult as there are still sections of assembly that reflect original design decisions that the current maintainers are unfamiliar with. Hopefully, there can continue to be meaningful enhancement and improvement in GSplus, but someday it may need to be replaced with another core, or perhaps a new emulator altogether.

We do what we can in the meantime.

Support and Development

This is an Open Source project that strives to achieve some level of professionalism and polish in the final product, but it is primarily the pet project of one person, so support is limited and development is not always at regular intervals. Still, the idea is to keep progressing, even if there are gaps and slowdowns in progress. Please feel free to get involved however you feel appropriate.

Getting Support

This project is hosted on GitHub at this page:

- ❖ <https://github.com/digarok/gsplus>

If you have a bug, please report it by going to the above page, clicking the “Issues” tab and submitting a “New Issue”.

There aren’t direct resources for support from the author, but the community is a great resource for answers to Apple IIGS and GSplus questions. Some places to check:

- ❖ comp.sys.apple2 newsgroup
- ❖ Apple II Enthusiasts FB group
- ❖ Apple IIGS Enthusiasts FB group
- ❖ Tweet @65816guy on Twitter

Don’t forget to check the Troubleshooting Appendices at the end of this manual for some common support issues.

Developer Contribution

If you would like to contribute to the project, head over to the GitHub page:

- ❖ <https://github.com/digarok/gsplus>

If you consider yourself non-technical, but would like to help with documentation, art assets, or other suggestions, feel free to use the “Issues” tab and submit a “New Issue”

If you would like to contribute code back to the GSplus, please create a personal fork on GitHub, commit your changes there, and submit a Pull Request to the project above.



Getting Started



GSplus is essentially an “old-school” command-line program. For some this is daunting, but this document makes an effort to calm any fears. Once you get it working, you are sure to have hours of fun exploring what the Apple IIGS has to offer.

Installing GSplus

- ❖ Download the appropriate package at <https://apple2.gs/plus>
- ❖ Install it somewhere on your system
 - Mac OSX - Open the DMG and drag GSplus into Applications
 - Linux - Uncompress to the folder of your choice
 - Windows - Unzip to the folder of your choice
- ❖ Install an Apple IIGS ROM
 - Download the firmware for either a ROM1 or a ROM3 Apple IIGS
 - ftp://ftp.apple.asimov.net/pub/apple_II/emulators/rom_images/
 - or*
 - <https://apple2.gs/downloads/ROM1> (128KB)
 - <https://apple2.gs/downloads/ROM3> (256KB)
 - If you have a real Apple IIGS, you can look up ways to dump your ROM to a file and transfer it to a PC.
 - Put the ROM image in the same folder where you put the emulator and it will search for: "ROM", "ROM1", "ROM3", in that order
 - or*
 - Put it in any folder and set the path in config.txt (see Configuration Files).
 - Example
 - `g_cfg_rom_path = ../roms/gsrom03`
- ❖ NOTE: You can also set the path of your Apple IIGS ROM file while GSplus is running by hitting F4 (see Configuration menu)

Launching GSplus

You start GSplus from the command line. You can launch it with no options once your system is configured the way you like. You can also set up a variety of configuration files and launcher shortcuts to have a variety of pre-configured Apple IIGS machines. One could set up icons on their desktop to launch straight into specific titles like Arkanoid, Zany Golf, or Oregon Trail.

- ❖ Launch GSplus by opening a shell in its directory and executing it
 - Mac OSX `./gsplus`
 - Linux `./gsplus`
 - Windows `gsplus.exe`

When GSplus starts you should see informational messages logged to the console window and it will then open a new window to show the Apple IIGS display.

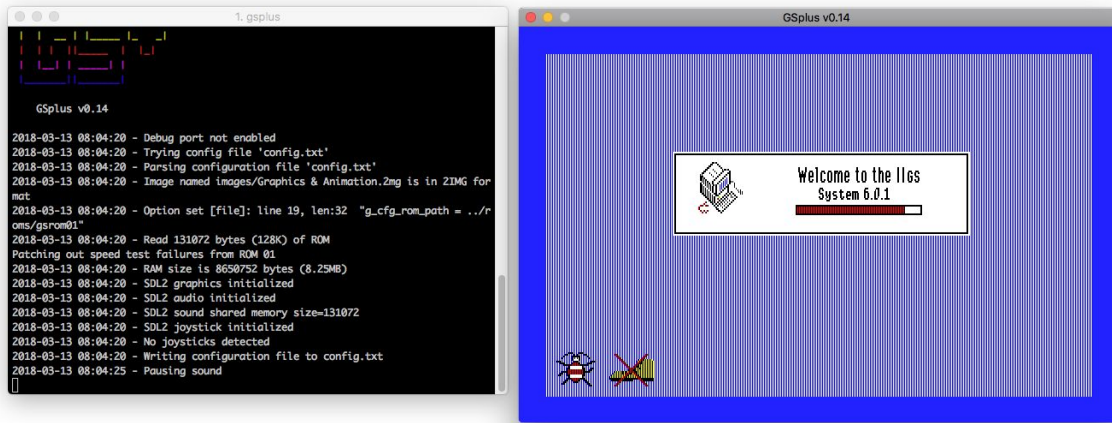


Figure 2-1

Booting GSplus emulator (left: console log - right: Apple IIGS display)

If any errors occur and/or the emulator window doesn't open, see the Troubleshooting appendix.

Emulator Configuration

As the emulator represents an entire computer with a variety of peripherals and states, we need some way to describe what our computer configuration looks like.

GSplus has three places where emulator configuration can happen. This may sound like a confusing design, but it makes a lot of sense as they represent different stages, or points in time, where you would want to control configuration:

Table 2-1

GSplus configuration layers

Config	Read	Notes
config.txt	At launch	File updates while running with any changes from below
CLI options	After config.txt	These override any options set above and are saved back
Menu (F4)	IIGS running	Saves changes to config.txt when you exit

Configuration Files

A huge part of how GSplus works is set up through a configuration file. This file describes things about the Apple IIGS you are running, such as how much memory the system has, what disks are inserted in the drives, and how fast the machine is running.

```
# GSplus configuration file version 0.14

s5d1 = images/arkanoid.2mg
s5d2 =

s6d1 =
s6d2 =

s7d1 =

g_cfg_rom_path = ../roms/gsrom01
g_limit_speed = 0

bram1[00] = 00 00 00 01 00 00 0d 06 02 01 01 00 01 00 00 00
bram1[10] = 00 00 07 06 02 01 01 00 01 00 0f 00 00 00 05 06
           ... skip ...
bram3[e0] = ff ff ff ff ff ff ff ff ff ff ff ff ff ff ff
bram3[f0] = ff ff ff ff ff ff ff ff ff ff ff ff 7b 49 d1 e3
```

Figure 2-2

Example config.txt file

In the example above, you can see we have one disk image (arkanoid.2mg) inserted in Slot #5, Disk #1 (s5d1). The other slots are empty. Below the drives is `g_cfg_rom_path` showing where our Apple IIGS firmware file is located, and `g_limit_speed` set to 0 which allows the emulator to run as fast as possible. Finally there's a series of bytes representing the computer's battery backed RAM state (bram) for both a ROM1 and ROM3 Apple IIGS.

Table 2-2

GSplus Drive/Slot mapping

Slot	Drives	Drive Type	Disk Size
5	x2	3.5" Floppy Disk	800KB
6	x2	5.25" Floppy Disk	140KB
7	x11	Hard Drive	Up to 32MB ProDOS (more with HFS)

Typically the sizes should exactly match when inserting disk images in slots 5 and 6, but the emulator will try to mount "wrong" sizes and do its best.

As the table shows, you can mount up to 11 hard drives, going from s7d1 to s7d11.

Default configuration files

By default, GSplus uses a file called `config.txt`. Very original. When you launch GSplus without any options, it looks for `config.txt`, `config.gsp`, and `.config.gsp` in the current directory. If it can't find any of those, it attempts to create `config.txt` and launch using that for its configuration.

While the emulator is running, it will save any changes to the configuration file if you make changes to the emulated Apple IIGS, like inserting or ejecting disks, changing Control Panel settings, or changing emulator speed. It also tries to save the configuration file when you shut down the emulator.

Multiple custom configuration files

One of the great features of GSplus is the ability to specify which configuration file to use when launching. In this way, you can have many pre-configured systems that are ready to boot up right into Karateka, Oregon Trail, or Arkanoid II. You could create shortcuts on your hosts desktop configured to launch individual pieces of software you want to boot right into to play. Or you could use this with a frontend emulation launcher.

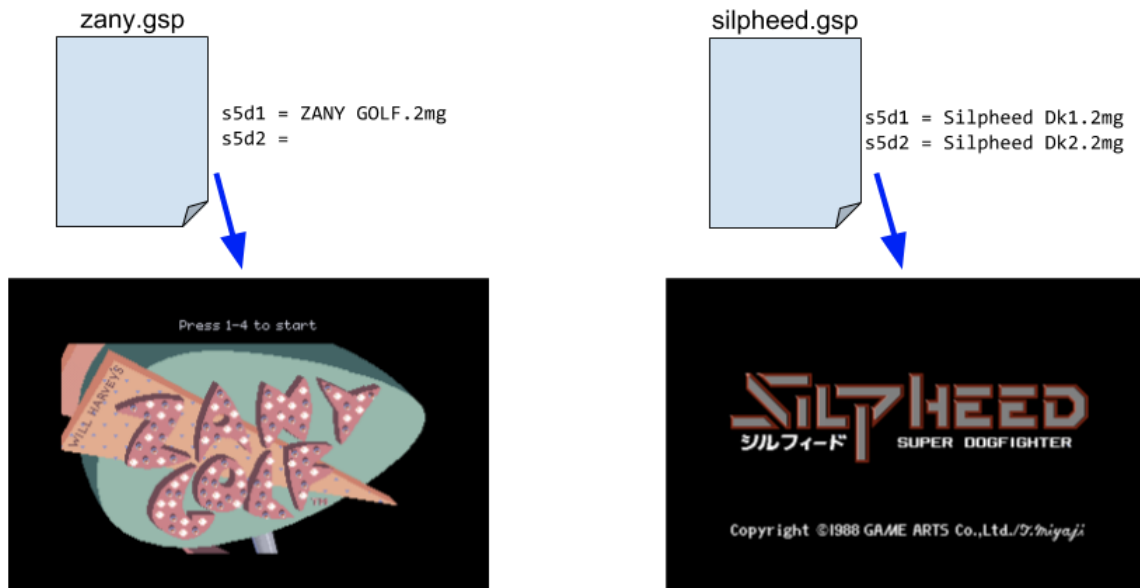


Figure 2-3

Using more than one config

In the example above, the only real difference between the two systems is that they have different disks mounted in slot 5.

It also is possible to pass a disk image as the final command-line argument when launching GSplus, and it will try to mount it. That's not a very sophisticated approach but can be useful for quick testing.

The multiple configuration file scenario is very helpful for many reasons. For most 8-bit Apple II games you want the Control Panel (and therefore BRAM) set to boot slot 6, whereas for most

16-bit Apple IIGS software you want the system to boot from slot 5 for a 3.5” disk or slot 7 for a hard drive. This simplifies things by having the correct disk inserted and configured to boot. But is also includes other nice customization settings like your mouse speed and border color. It is common to run many games, especially older 8-bit titles, with a black border.

The only difference when launching with a custom config file is the `-config` argument.

Example:

```
./gsplus -config myconfig.txt
```

Configuration Menu

While the emulator is running, you still need to be able to update the machine’s state. One classic example is that you need to be able to “insert” or “eject” floppy disks from time to time. To do this, GSplus provides a configuration menu that is accessible at any time by hitting **F4**.

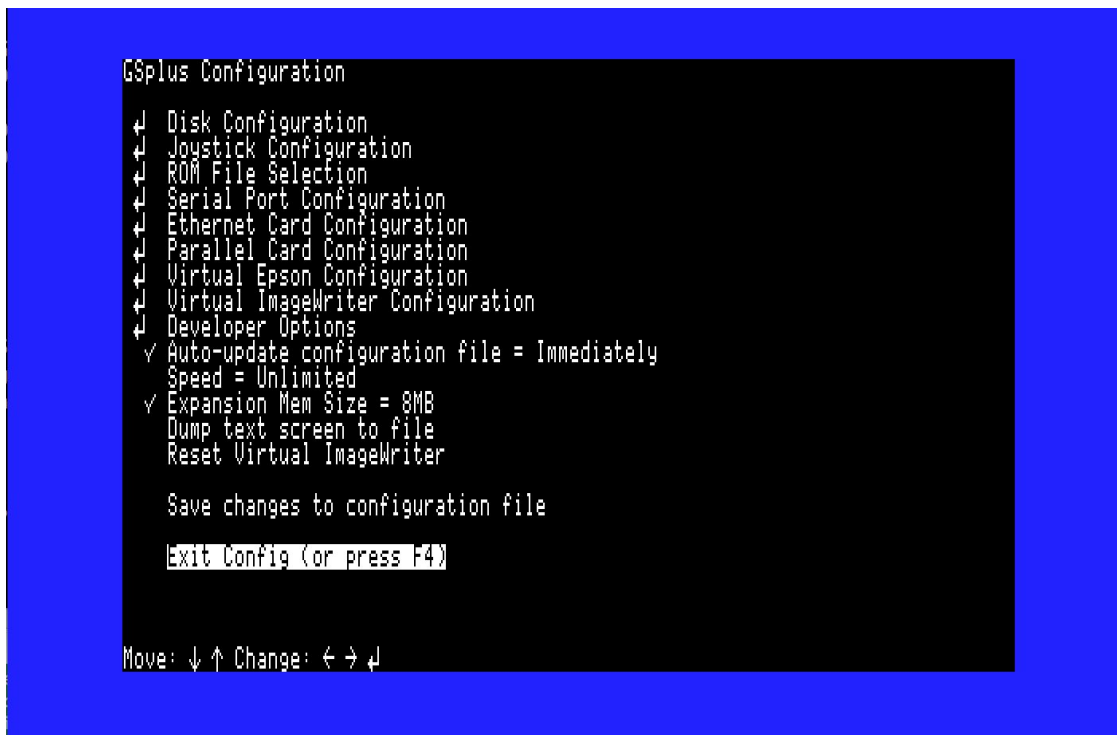


Figure 2-4

The “<F4>” configuration menu

The most important options are:

- ❖ Disk Configuration - use this to “insert” or “eject” floppy disk or hard drive images
- ❖ ROM File Selection - if this isn’t pointing to a valid ROM 1 or ROM 3 image, the emulator won’t boot (see the Troubleshooting appendix for more help with booting)
- ❖ Memory - set the amount of RAM in the emulated IIGS (requires restart)
- ❖ Speed - toggle between 1MHz, 2.8MHz, 8MHz and “unlimited” speed (also right mouse)

By default, any changes made in this menu are save to your configuration file when you exit this menu.

Command Line Options

When you start the emulator, you can send a large number of options for configuration via the command line. We've already described the scenario of using `-config` option to load a specific configuration, but there are many other options that you can use to launch the emulator into a specific configuration, regardless of settings in the configuration file.

You can always look at the latest list of options in your version of GSplus by running:

```
./gsplus -?
```

This shows a short help message followed by all the command line options available. Some command line options enable a feature, like `-fullscreen` tells the emulator to try to launch in fullscreen mode. Some command line options require a second argument, like `-joystick_number` must have the number of the joystick in your host PC which you want to use as the second argument.

Here's a more complex launch example:

```
./gsplus -config robotron.txt -joystick_number 1 -fullscreen -scanline 30
```

This tells GSplus to:

- ❖ Launch with the "robotron.txt" configuration file
- ❖ Use the second joystick (will still default to the first two axes/buttons)
- ❖ Enable fullscreen on launch
- ❖ Turn on simulated CRT scanlines at 30% darkness

You could also open multiple terminals and launch multiple instances of GSplus. Just be aware that you should specify different configuration files (or be running in different directories with a local config.txt). You technically can run two instances with the same config file/directory, but making changes in one machine will make it out of sync with the other machine and could break things.

Remember that any settings passed as command line options will override those values in the configuration file, and be saved back to the configuration file.

A full list of command line options is available in "Appendix A - Full list of configuration options" at the end of this manual.



Input Devices



GSplus will generally work out-of-the-box with your modern computer's input devices. However, there are a variety of ways to adjust how they function. One note: this documentation is specifically covers the SDL2 versions of GSplus. Older versions are not supported.

Mouse

The Apple IIGS used an ADB (Apple Desktop Bus) mouse that had a fairly “modern” interface for the time. GSplus will automatically translate your system's mouse into the ADB mouse and should work quite well with traditional GS/OS based software.

Compatibility issues do arise. One problem is related to mouse scaling under certain titles, particularly some 8-bit mouse programs.

Some tips in particular that may help you with mouse movement:

- ❖ Visit the Apple IIGS Control Panel (OpenApple-Control-Escape) and set the mouse speed to “Fast”
- ❖ Run the emulator in fullscreen (especially for games where you don't want the mouse to be able to leave the focus area of the Apple IIGS)
- ❖ Toggle warp pointer with F8

Joystick

The Apple IIGS used the same joysticks as the earlier 8-bit Apple II computers. They are simple analog designs based on an NE558 timer chip. It was a quad-timer, so could be used to record up to 4 axes, meaning 4 paddles, or 2 joysticks. However, in reality most Apple II software only ever supported one joystick (of two axes and two buttons).

With that in mind, the current SDL2 driver only supports one joystick input device on your host PC, but you can remap the axes and buttons however you like.

On an Apple II, the 4 axes are referred to as “Paddles” 0 through 3. Games (mostly) expected a joystick to use Paddle(0) for the X axis and Paddle(1) for the Y axis.

By default, GSplus uses your first gamepad. Typically you don't need to make adjustments as most “console-style” USB dual-stick gamepads map their left stick to axis 0 and axis 1, which are the GSplus defaults.

Joystick Options

There are two types of options to set with GSplus: Controller Mappings and Tweaks

The controller mappings let you set how it maps your USB gamepad to the Apple II device. The available mappings (SDL2) are:

- ❖ Joystick number (which SDL2 gamepad device to use)
- ❖ Joystick x and y Axis
- ❖ Second joystick x and y axis (must be on the same gamepad)
- ❖ Joystick buttons 0 through 3

The controller tweaks affect how any controller behaves, outside of the gamepad driver.

The available tweaks are:

- ❖ Joystick emulation (use numpad or mouse if you don't have a gamepad installed)
- ❖ Joystick x and y axis scale factor (-30% to 30%)
- ❖ Joystick x and y axis trim factor (nudge left/right/up/down if controller is off-center)
- ❖ Swap paddles 0 and 1
- ❖ Invert axes (min becomes max)

You can set all of the tweaks in the Configuration Menu (**F4**), and some you can toggle via hotkey. Currently the mappings are only available as command line options. To use them, see “Appendix A - Full list of configuration options”.

Keyboard

The Apple IIGS used a well-beloved ADB keyboard and just like with the mouse, it should mostly work out-of-the-box with whatever keyboard you have on your host machine.

All of the US keyboard letters, numbers and symbols respond as one would expect, for the most part. The Apple IIGS had two primary modifier keys, “Open Apple” and “Closed Apple”, and the typical “Control” key. It also had a special “Reset” key at the top of the keyboard used to execute the soft and hard reset key combinations described below.

Apple IIGS Keyboard Map

On modern PCs we have two keyboard styles, Windows/Linux or Mac. The main differences being the modifier keys use different names. Here's how keys are mapped for the various implementations:

Table 3-1
Modifier key mapping

Apple IIGS	Windows	Linux	Mac OSX
Open-Apple	Alt	Alt	Command
Option	Win	Option/Tux	Option
Reset	F12	F12	F12

The Apple IIGS has a few special features that can be invoked by special modifier keyboard combinations. You can use the mappings above to figure out how to do this on your system.

Table 3-2
Apple IIGS special keyboard combinations

Input	Command	Description
Open-Apple-Control-Reset	Hard Reset	Should reboot the computer
Control-Reset	Reset	Break to monitor/reset vector
Open-Apple-Control-Escape	Control-Panel	Set Apple IIGS configuration options
Open-Apple-Option-Control-Reset	System Test	Run the built-in diagnostic firmware

Emulator Hotkeys

It is useful to be able to quickly change some emulator settings on-the-fly. Outside of using the Configuration Menu, there are also a variety of emulator hotkeys you can use.

F1	- Open Apple (Command)
F2	- Option
F3	- screenshot *
F4	- configuration menu - set lots of options here!
F5	- paste from host OS into emulator
F6	- toggle system speed (1, 2.8, 8, ∞ MHz)
Shift+ F6	- HALT2 (?)
F7	- toggle fast disk emulation
F8	- toggle warp pointer
F9	- invert paddles
Shift+ F9	- swap paddles
F10	- toggle a2vid_palette (?)
Shift+ F10	- toggle status line *
F11	- toggle fullscreen mode *
Shift+ F11	- toggle simulated scanlines *
F12	- hit RESET (e.g. - to do "Ctrl-Reset" on an Apple II, you hit "Ctrl-F12")

* only SDL2 versions

Note : The right mouse button also works as a hotkey to toggle emulator speeds.



Advanced Usage



This discusses some of the idiosyncrasies of GSplus and how to deal with them.

Debugging

There is a built in CPU debugger in GSplus but it is not very good, in my opinion. Further work is going on to improve this, and will likely be the focus of the next release.

Special Configuration Tips

- If you'd like a game/app to always use a certain border/text/bg color combination, set it in the Apple IIgs control-panel and exit to let it get saved to battery RAM (bram) which updates the config file. Then just copy that config file to a new name and always launch your app like that:
 - `./gsplus -config mygame_config.txt`
- This also works for setting the boot slot and system speed. So if your game is a 5.25" disk, set the boot slot to 6 in the Control-Panel, etc.
- Some demos/games/apps would auto-eject the 3.5" disk after loading, e.g.- *Modulae - By The FTA*. One way to be able to always immediately boot into the demo is to create your config file with the Modulae disk loaded in slot 5. Then on your host OS, set the config file to Read Only.

```
$ chmod -w modulae_config.txt
```

This is a bit of a work-around, but can be quite useful if you want to have pre-configured shortcuts for demos like this.
- You could potentially also achieve the above by passing in the name of your disk as the last argument to gsplus, as it always tries to mount the last command line argument as a disk.

Troubleshooting

Emulator won't start

- TIP!! If you are launching by double-clicking on the icon, then the first suggestion is to try launching it from the command line. There's a lot of debug output that could help you. It will even try to diagnose a lot of the following conditions.
- Screen full of "@@@@@":

- No ROM file. You must supply your own IIgs ROM file. If you get a screen full of “@@@” then you are missing a ROM. See the Installing GSplus section of Chapter 2 - Getting Started
- Makes a beep noise and shows a “*” prompt
 - No boot disk - You either need to mount a volume (F4), or if it is mounted, you need to go to the Apple IIgs Control-Panel to set the boot slot.
 - Tip, from the “*” prompt, you can try to boot any slot by typing “Cx00G” where x is the slot number:
* C500G <- Boot slot 5
- Fails silently.
 - This could be many things. Try creating or copying a “config.txt” file in the directory you are launching from to be sure it isn’t having a problem finding or auto-creating it.
 - It’s possible it’s caused by a graphical, joystick, or audio problem. You might try disabling audio, or joysticks.

Mouse movement is off

- Try running in full-screen (F11)
- Set the mouse speed to Fast in the Apple IIgs Control-Panel
- Turn on Warp Pointer (F8)

Computer is too slow/fast

- Set the emulator speed in the config menu (F4) or use right-mouse button
- Tip!! Make sure that the Apple IIgs Control-Panel speed setting is set to Fast, otherwise you will run at 1MHz, no matter how fast you set the emulator.

Hotkeys (Fullscreen) not working on my Mac

- Make sure you are pressing the “fn” key if it’s an Fn key on a Mac keyboard.
- I also have to hit “control” with most of the Fn key shortcuts to keep OSX from thinking it’s one of its commands.
 - Normal computer fullscreen = “F11”
 - Stupid Mac fullscreen = “fn-Ctrl-F11”

Windows is slow / shows weird paths.

- Windows version uses Cygwin with Mingw SDL2 libraries and suffers from some “unique” problems. While it is functional, the hope is that there will be time to provide a native build in the future.

Border effects look wrong

- They are.



Appendix A



This is the full list of configuration options currently supported by GSplus command line. Some can also be changed/toggled by hotkey or via the Configuration Menu (F4), as represented here

Full list of configuration options

CLI Parm	Option Name	Config Menu?	HotKey	Values	Category
-audio	Audio enable option	No		0-?	Audio
-arate	Audio rate	No		8000-48000	Audio
-fullscreen	Fullscreen	No	F11		Display
-highdpi	High DPI display	No			Display
-borderless	Borderless window	No			Display
-resizeable	Resizeable window	No			Display
-skip	Screen redraw skip frames	No		0-?	Display
-dhr140	Use simple d hires color map	No			Display
-bw	Force B/W hires modes	No			Display
-scanline	Enable scanline simulation %	No	shift-F11	0-100	Display
-x	Create Window a X	No		0-?	Display
-y	Create Window a Y	No		0-?	Display
-joy	Joystick/Gamepad to use	No		0-?	Input
-joy_x	X axis number	No		0-?	Input
-joy_y	Y axis number	No		0-?	Input
-joy_x2	X axis number (joy2)	No		0-?	Input
-joy_y2	Y axis number (joy2)	No		0-?	Input
-joy_b0	First button number	No		0-?	Input
-joy_b1	Second button number	No		0-?	Input
-joy_b2	Third button number	No		0-?	Input
-joy_b3	Fourth button number	No		0-?	Input
-enet	Enable Ethernet	Yes		0-?	Networking
-badrd	Halt on bad read	No			System
-noignbadacc	No ignoring bad RAM access	No			System
-noignhalt	No ignoring code red halts	No			System
-mem	Expansion Mem Size (RAM)	Yes		0-14680064	System
-config	Use specific configuration file	No		{filename}	System
-ssdir	Screenshot directory	No		{path}	System



Appendix B



Here are some documentation resources on the Apple IIGS computer itself. This list is not an endorsement, just an effort to help those who may not have historical experience with the computer. You can find these online as used books or PDF scans.

Apple IIGS Documentation

For the new user:

- ❖ Apple IIGS Owners Guide
- ❖ Apple IIGS System Disk User's Guide
- ❖ Apple IIGS System 6 User's Reference

For those wanting to learn more about the system and hardware:

- ❖ Apple IIGS Hardware Reference (2nd Ed.)
- ❖ Technical Introduction to the Apple IIGS

For those wanting to quickly play around with BASIC programming:

- ❖ A Touch of Applesoft BASIC

For those wanting to become developers:

- ❖ Programmer's Introduction to the Apple IIGS
- ❖ Apple IIGS GS/OS Reference
- ❖ Apple IIGS Toolbox Reference - Volume 1
- ❖ Apple IIGS Toolbox Reference - Volume 2
- ❖ Apple IIGS Toolbox Reference - Volume 3
- ❖ Le IIGS Épluché
- ❖ Programming the 65816 (Including the 6502, 65C02 and 65802)
- ❖ Compute!'s Apple IIGS machine language for beginners
- ... and many many more

Online there are many references. GSplus is hosted at <https://apple2.gs> and the same site includes a lot of programming information in the "Library" section.

Good luck and have fun!



Appendix C



GSplus represents a dream of an improved emulator that helps our Apple II developers as well as those who simply want to play a quick game of Marble Madness on their modern PC. Programming the Apple II and playing games on it is quite fun. Making an emulator for it is decidedly less so. Sometimes the work seems thankless and unfixable. It is with a sincere sense of gratitude that I thank the following people who have made this all possible through their own encouragement, contribution and energy.

Credits and Thanks

Kent Dickey - Author of KEGS, without whom, the Apple IIGS emulation and enthusiast scenes would be only a miniscule portion of their current stature. There are other emulators, but this one has always been available for a variety of platforms and GSplus is 100% derivative work of Mr. Dickey's efforts. Thanks!

GSPort Contributors - GSplus started before GSPort as a direct fork of KEGS, but a few years ago the decision was made to start a new fork from GSPort as a way to incorporate their enhancements. Initially, many things had to be broken in GSPort to get builds working and integrate legacy GSplus code. However, once the dust settled, it was clear there were a ton of configuration enhancements that could be useful to all emulation users. We are lucky to have the work of the GSPort contributors. Thanks!

Kelvin Sherlock - In the early days of making GSplus pseudo-public, one contributor was brave enough to deal with a poor branching structure and a poorly maintained repository and contribute some incredible features to GSplus. Mr. Sherlock has helped motivate and focus direction by getting very literal with the idea of modern features, adding portions like the disk image "drag-n-drop" and host FST. It should also be recognized that he also does a lot of work on development tools (MPW) for the Apple IIGS community that is also incredibly important. Thanks!

Jesse Blue, Antoine Vignau, Olivier Zardini, and John Brooks - I am lucky to have a lot of conversations with the best programmers in the Apple IIGS community. These people also work tirelessly on tools and products to support the Apple II community and showcase the platform. When motivation is lacking, these folks always appear and blow my mind with their behind-the-scenes projects. It's a gentle reminder that we are all working hard, and a lot of our work has the side-effect of enabling our comrades in the community. Thanks!

Chris Shepherd - For helping me pick up assembly language way back when. Thanks!

My Family - I cannot believe the sacrifice you've given in the cause of supporting this work. You are incredible and your love keeps me going.

You - I'm truly thankful that you even have an interest in the Apple IIGS. It was a lot of fun. I think it was largely misunderstood, oddly marketed, and intended to be forgotten by Apple Computer in the 1990s. The platform may have faced a dead-end with the 65xx CPU eventually, but there was so much life left that was never explored by its corporate overlords. Genuinely, it was the last of the ultimate hardware and software hacking platforms before personal computing became something different. People are still creating ethernet cards, flash drives, CPU accelerators, sound cards, and specialized preservation hardware. Likewise, there are at least a few new games a year and tons of utilities and smaller programs being released at a constant pace. Maybe this is a short-live renaissance, and that is why I am thankful for your interest in the Apple IIGS. As long as we remember it, it exists, and maybe a small part of me continues to exist as well. Thanks.

// Infinitum