A7800

Version 4.0

http://7800.8bitdev.org/index.php/A7800_Emulator

Table of Contents

What is it? 3

Quick and easy portable install for Windows 4

Starting an Atari 7800 console system 5

Temperatures.......... 6

Standard Console 7

High Score Cartridge 7

XM Expansion Module 7

System Main Menu 9

Load ROM File 9

Controller Selection 11

Console & Controller Inputs.......... 12

Analog Controls.......... 14

Crosshair Options.......... 14

Other mappings 15

Load & Save States.......... 16

Difficulty Switches.......... 17

Video and Audio Controls 18

Video GDI 20

Video BGFX.......... 20

HLSL.......... 23

Hardware & Information.......... 31

Debugger.......... 32

FAQs 34

What is it?

A7800 is a fork of the MAME Atari 7800 driver, with several enhancements added:

- Support for emulation of Proline Joysticks, VCS Joysticks, Lightguns, Paddles, Driving Controllers, Keypads, Trak-Balls, Amiga Mice, and ST Mice.
- Maria DMA timing has been improved further, with the addition of accurate DMA hole penalties.
- A bug in the existing RIOT emulation has been fixed.
- POKEY sound emulation improvements.
- SALLY (CPU) and MARIA (Graphics chip) performance adjustments.
- Color palette selection and enhanced screen options.
- Audio indication of no ROM loaded silenced.
- BIOS file(s) no longer required and made optional.
- Streamlined menus, updated hotkeys, and remapped defaults, providing a more Atari 7800-centric focus.
- Implementation of XM control registers updated.

NOTE: While there are exceptions and exclusions as well as new inclusions, for the most part, MAME compatibility and syntax has been maintained, to allow for the reuse of MAME configuration files and front-ends.

Quick and easy portable install for Windows

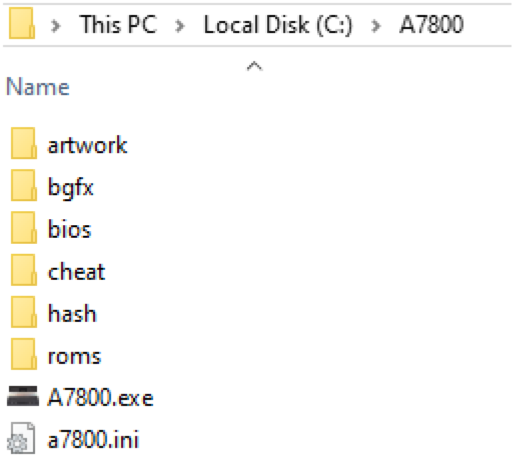
(For OS X and Linux ports see the A7800 Emulator Home Page)
http://7800.8bitdev.org/index.php/A7800_Emulator

1. Download the A7800 emulator and unzip to a new/clean location [I.E. *C:\A7800*].

2. [OPTIONAL] Place an obtained (merged) bios file(s) archive for the Atari 7800 under the same location within a bios subfolder [I.E. *C:\A7800\bios\a7800.zip*].

3. Place ROM files into a roms subfolder [I.E. *C:\A7800\roms*] for convenience. Otherwise, navigation to the ROMs location will be required while A7800 is running.

After the three steps are performed, the folder should appear similar to the below capture:



› This PC › Local Disk (C:) › A7800

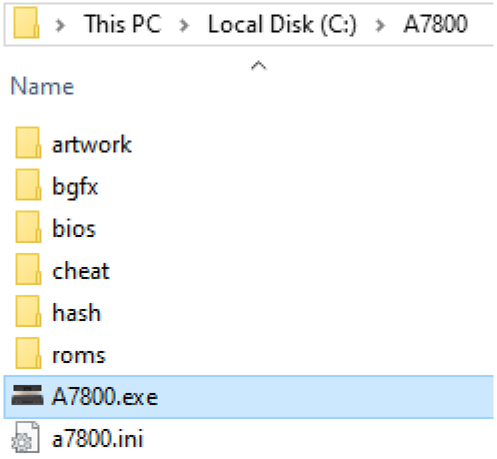
Name

artwork
bgfx
bios
cheat
hash
roms
A7800.exe
a7800.ini

<u>NOTE</u>: The current MAME convention of placing the system bios files in the 'roms' folder, with game and utility ROMs placed in a 'software' subfolder divided by system-driver name, may be followed alternatively.

Starting an Atari 7800 console system

Run the A7800 executable:

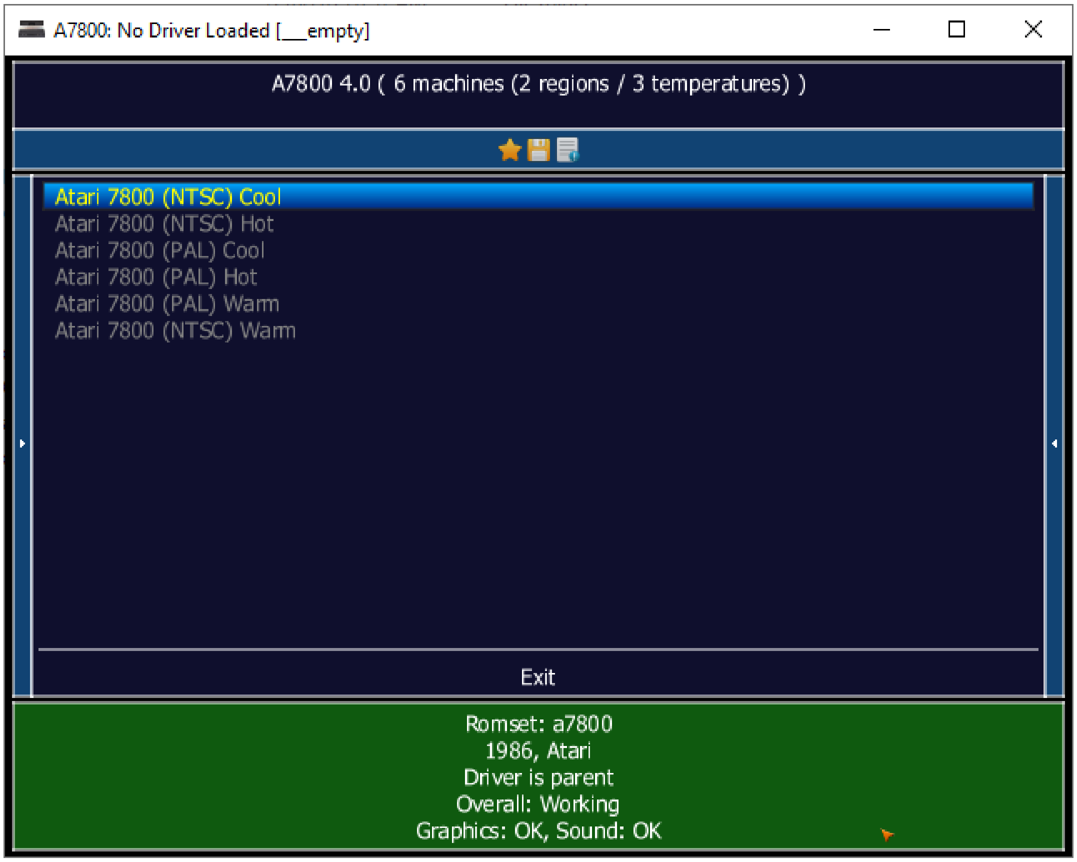


This PC › Local Disk (C:) › A7800

Name

artwork
bgfx
bios
cheat
hash
roms
A7800.exe
a7800.ini

Choose a console by double clicking with mouse pointer or highlighting with the arrow keys and pressing 'Enter', either NTSC or PAL, and the console temperature condition: Cold, Hot, or Warm. System temperature impacts the color palette:



A7800: No Driver Loaded [__empty]

A7800 4.0 (6 machines (2 regions / 3 temperatures))

Atari 7800 (NTSC) Cool
Atari 7800 (NTSC) Hot
Atari 7800 (PAL) Cool
Atari 7800 (PAL) Hot
Atari 7800 (PAL) Warm
Atari 7800 (NTSC) Warm

Exit

Romset: a7800
1986, Atari
Driver is parent
Overall: Working
Graphics: OK, Sound: OK

Temperatures

Cool: Hue Separation of ~25.7 degrees. This is the 'factory' setting and instructed method, per the *CPS 7800 Diagnostic Test Cartridge*, to calibrate the system's color pot located on the motherboard just below the cartridge slot. This seems typically to have been set within a minute or two of a system being powered on, as many factory set consoles appear to drift higher in hue separation as the system continues to warm up. Some early titles have their colors represented best at this setting such as *Galaga* and *Joust*. Titles that have their colors best showcased under "Cool" look their worst under "Hot".

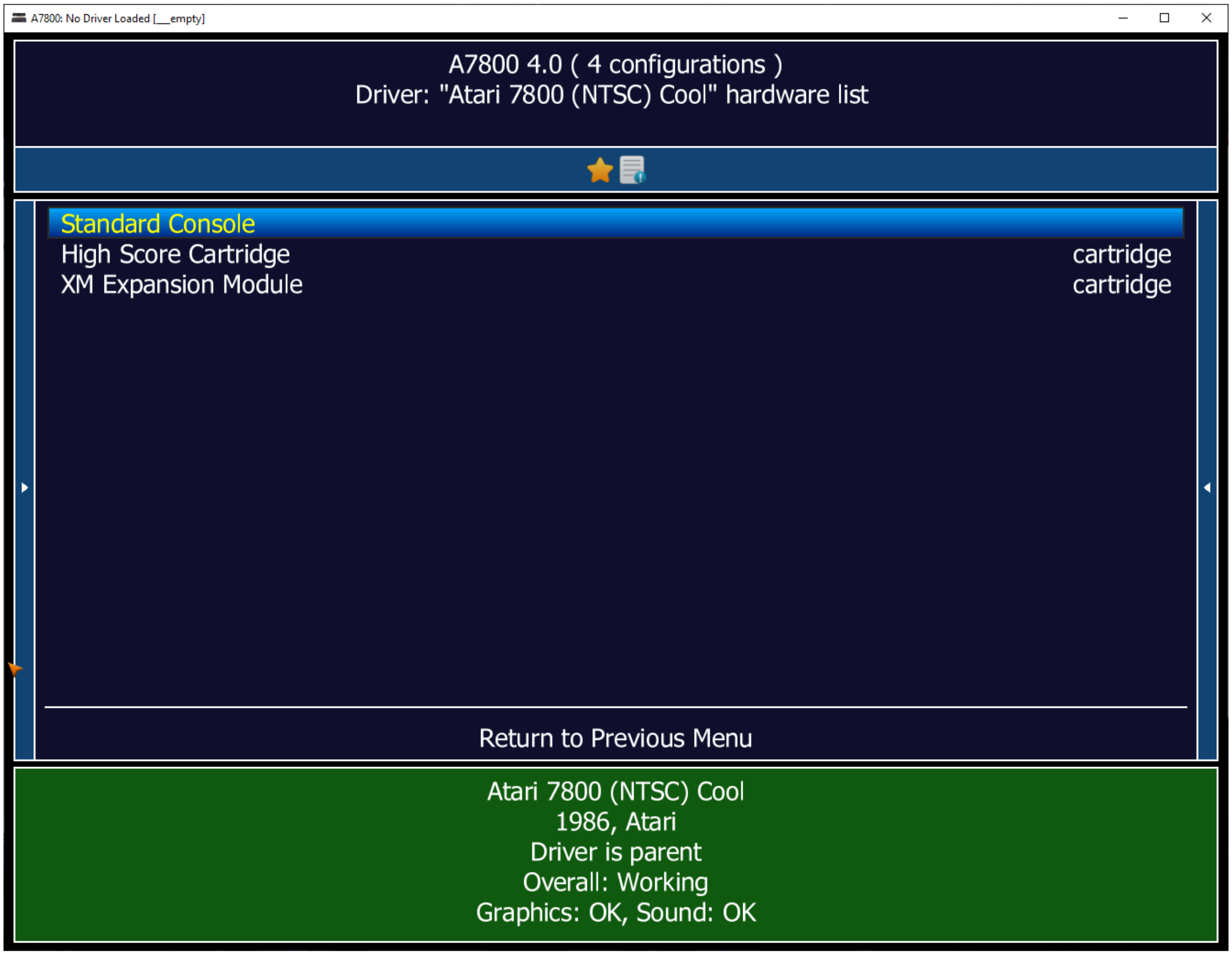
Hot: Hue Separation of ~27.7 degrees. This is the 'typical' setting often experience after a console has been running continuously for 15-30 minutes. Most late release titles such as *Midnight Mutants* and *Ikari Warriors*, as well as the earlier released *Choplifter,* display optimal colors when the palette is provisioned this way. Game colors looking best "Hot" have the least favorable appearance under "Cool".

Warm: Hue Separation of ~26.7 degrees. This is the 'ideal' setting as it best matches official documentation on the color order for the console, including a full 256 unique colors palette for the system. It allows for some system warm-up, resulting in several games yielding colors neither at their best or worst. However, other games do yield best color results with a "Warm" palette. *Scrapyard Dog*, for example, results in a fence and sky having the best balance of colors. Some games, such as *Dig Dug,* look excellent regardless of temperature.

For more information regarding colors, see the 7800 Color Documentation page located on the 7800.8bitdev.org site:

http://7800.8bitdev.org/index.php/Atari_7800_Color_Documentation

Select a console configuration with mouse pointer and double clicking on selection, or highlight with the arrow keys and pressing 'Enter':



A7800: No Driver Loaded [__empty]

A7800 4.0 (4 configurations)
Driver: "Atari 7800 (NTSC) Cool" hardware list

Atari 7800 (NTSC) Cool
1986, Atari
Driver is parent
Overall: Working
Graphics: OK, Sound: OK

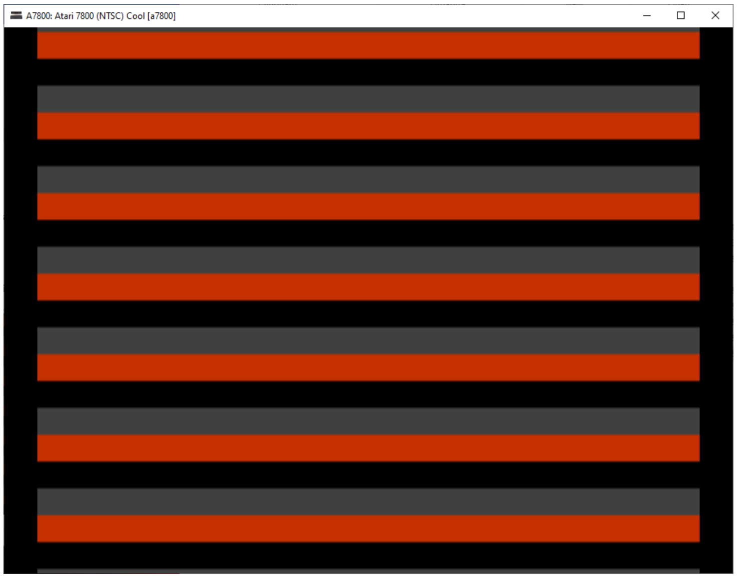
Standard Console: Base system with no add-on hardware present.

High Score Cartridge: Base system with high score saving cartridge pass-thru.

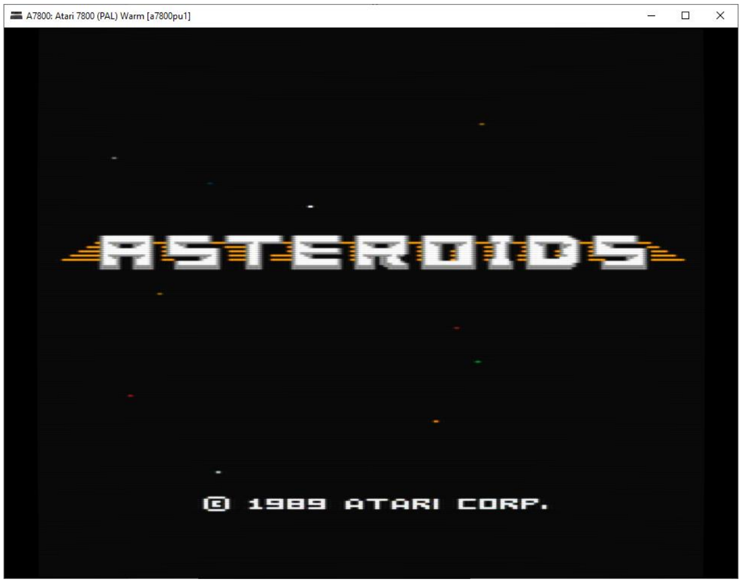
XM Expansion Module: Base system with High Score Cartridge imitation saving, added memory, POKEY and YM sound chips.

An Atari 7800 console is now running.

For NTSC consoles, it means the console is turned on with no (game) program ROM present, resulting in the following screen:

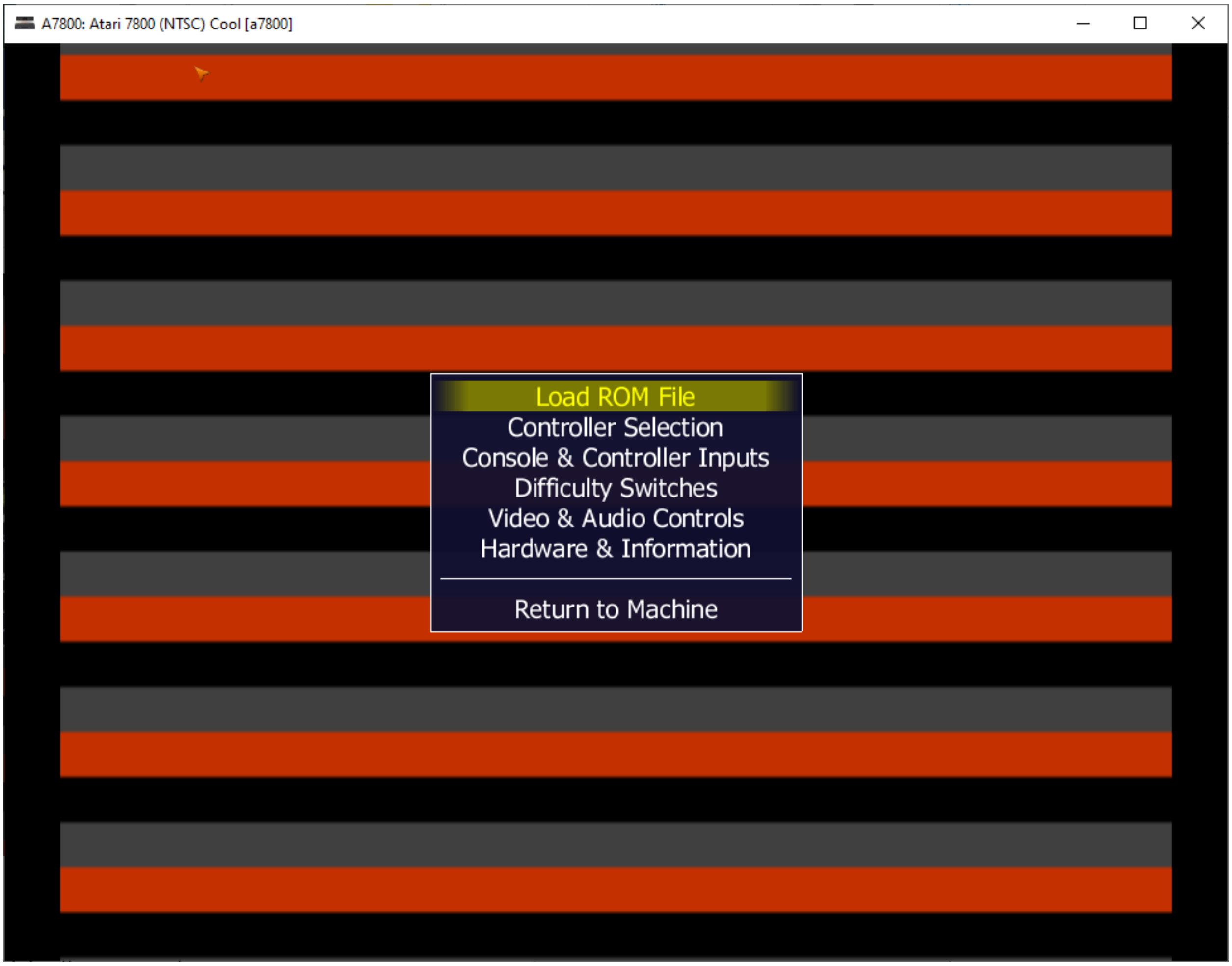


PAL consoles have Asteroids built-in and the game runs automatically when no cartridge is present and the console is powered on:



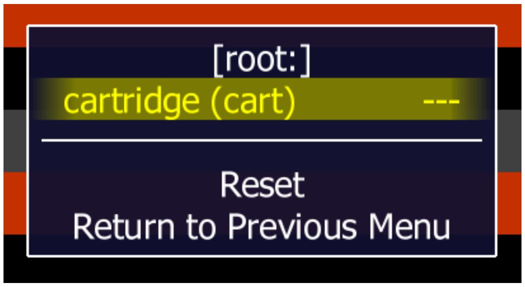
System Main Menu

While a system is running, the main menu is accessed by pressing the ‘Tab’ key:

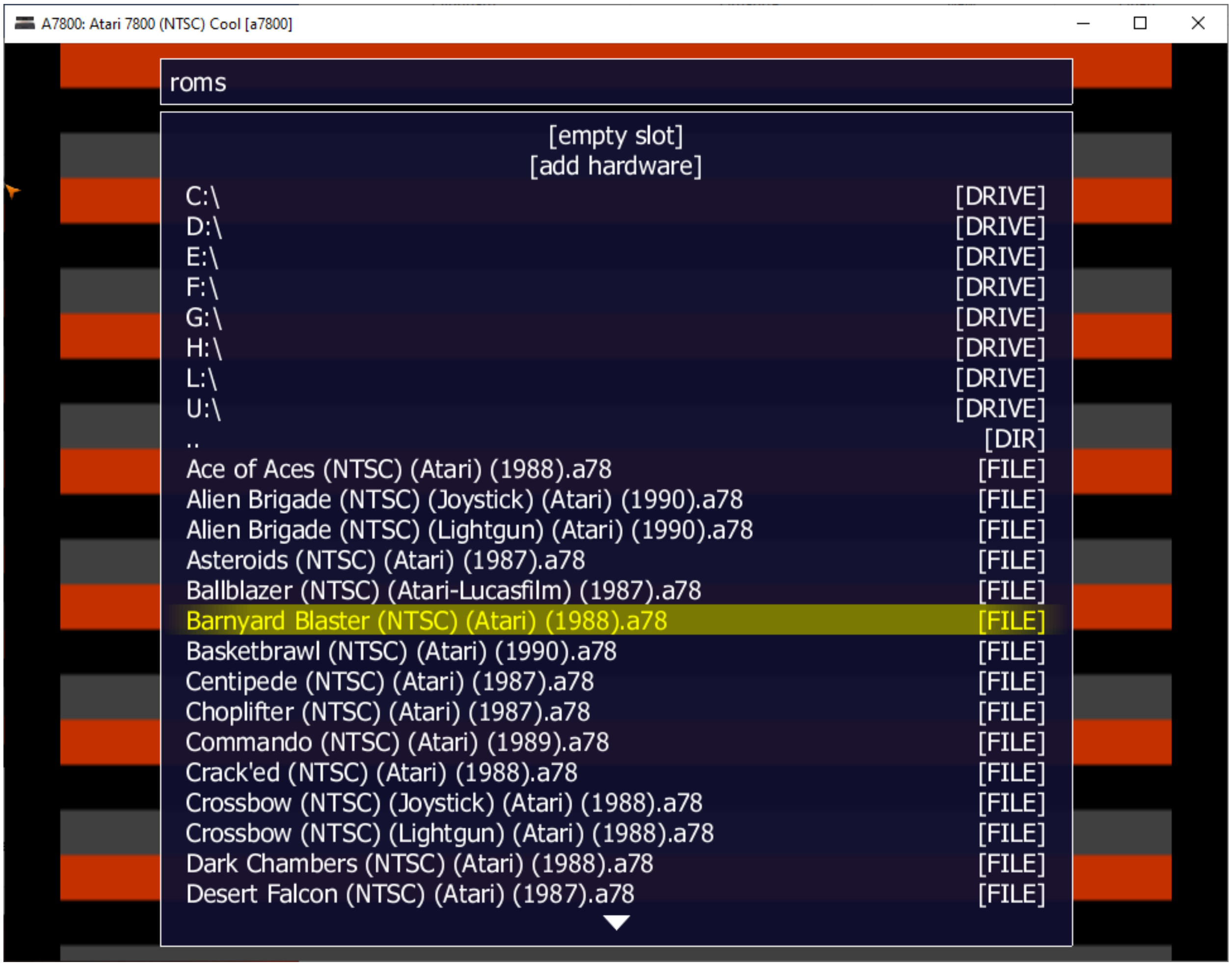


Load ROM File: Select a ROM file to run under the A7800 emulator. Highlight the option with arrow keys and press 'Enter', or double click with mouse pointer.

Perform the same respecting the available "cartridge (cart)" slot that appears:

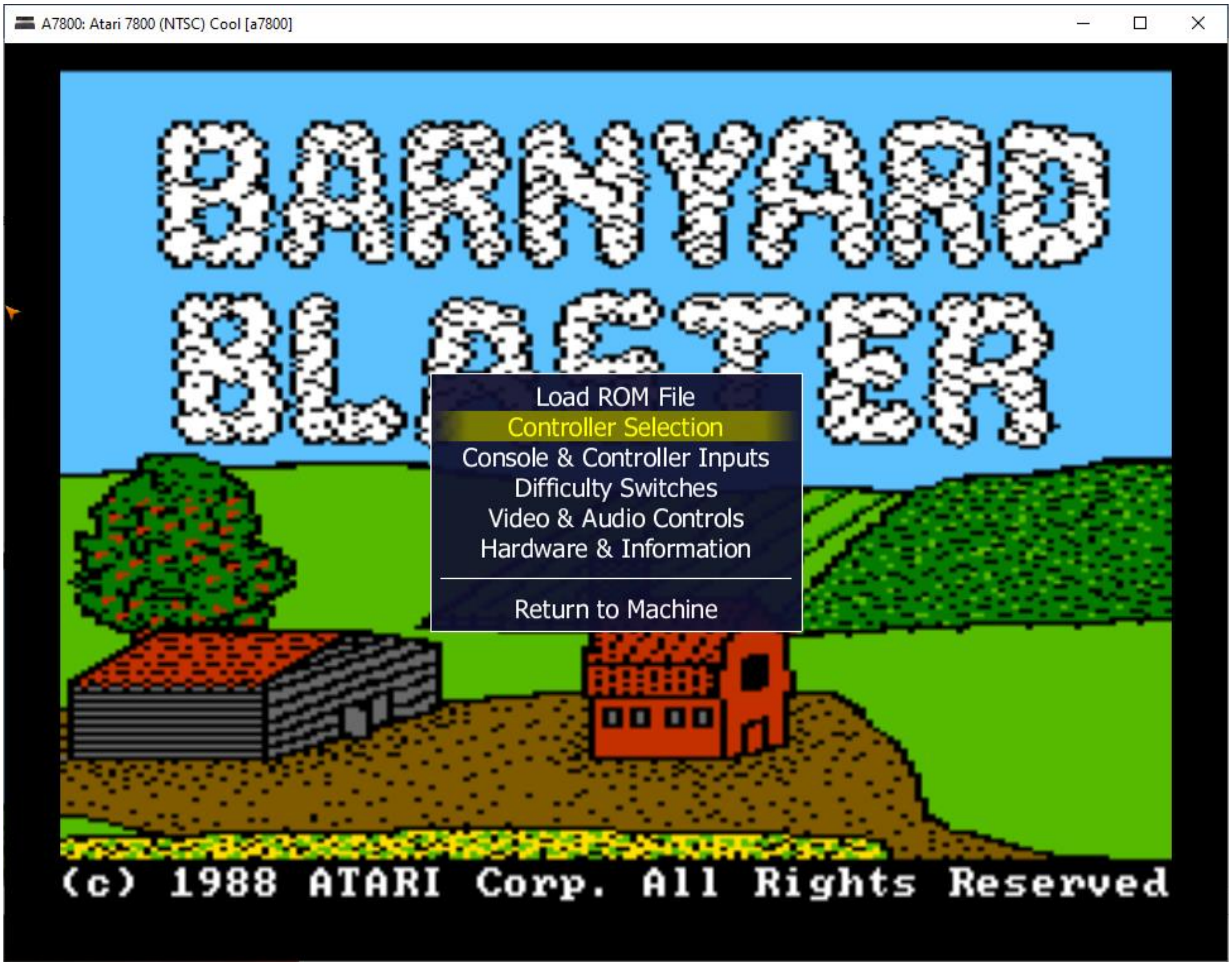


Choose a ROM from the displayed "roms" subfolder on the screen:



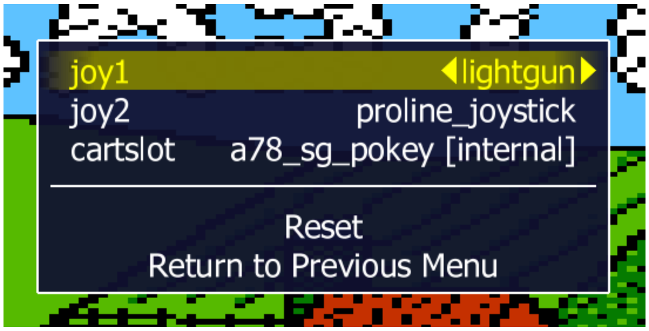
```
C:\                                                  [DRIVE]
D:\                                                  [DRIVE]
E:\                                                  [DRIVE]
F:\                                                  [DRIVE]
G:\                                                  [DRIVE]
H:\                                                  [DRIVE]
L:\                                                  [DRIVE]
U:\                                                  [DRIVE]
..                                                     [DIR]
Ace of Aces (NTSC) (Atari) (1988).a78                 [FILE]
Alien Brigade (NTSC) (Joystick) (Atari) (1990).a78    [FILE]
Alien Brigade (NTSC) (Lightgun) (Atari) (1990).a78    [FILE]
Asteroids (NTSC) (Atari) (1987).a78                   [FILE]
Ballblazer (NTSC) (Atari-Lucasfilm) (1987).a78        [FILE]
Barnyard Blaster (NTSC) (Atari) (1988).a78            [FILE]
Basketbrawl (NTSC) (Atari) (1990).a78                 [FILE]
Centipede (NTSC) (Atari) (1987).a78                   [FILE]
Choplifter (NTSC) (Atari) (1987).a78                  [FILE]
Commando (NTSC) (Atari) (1989).a78                    [FILE]
Crack'ed (NTSC) (Atari) (1988).a78                    [FILE]
Crossbow (NTSC) (Joystick) (Atari) (1988).a78         [FILE]
Crossbow (NTSC) (Lightgun) (Atari) (1988).a78         [FILE]
Dark Chambers (NTSC) (Atari) (1988).a78               [FILE]
Desert Falcon (NTSC) (Atari) (1987).a78               [FILE]
```

Otherwise, navigate to the location of where the ROMs are stored. ROMs should have headers in proper *.a78 format.



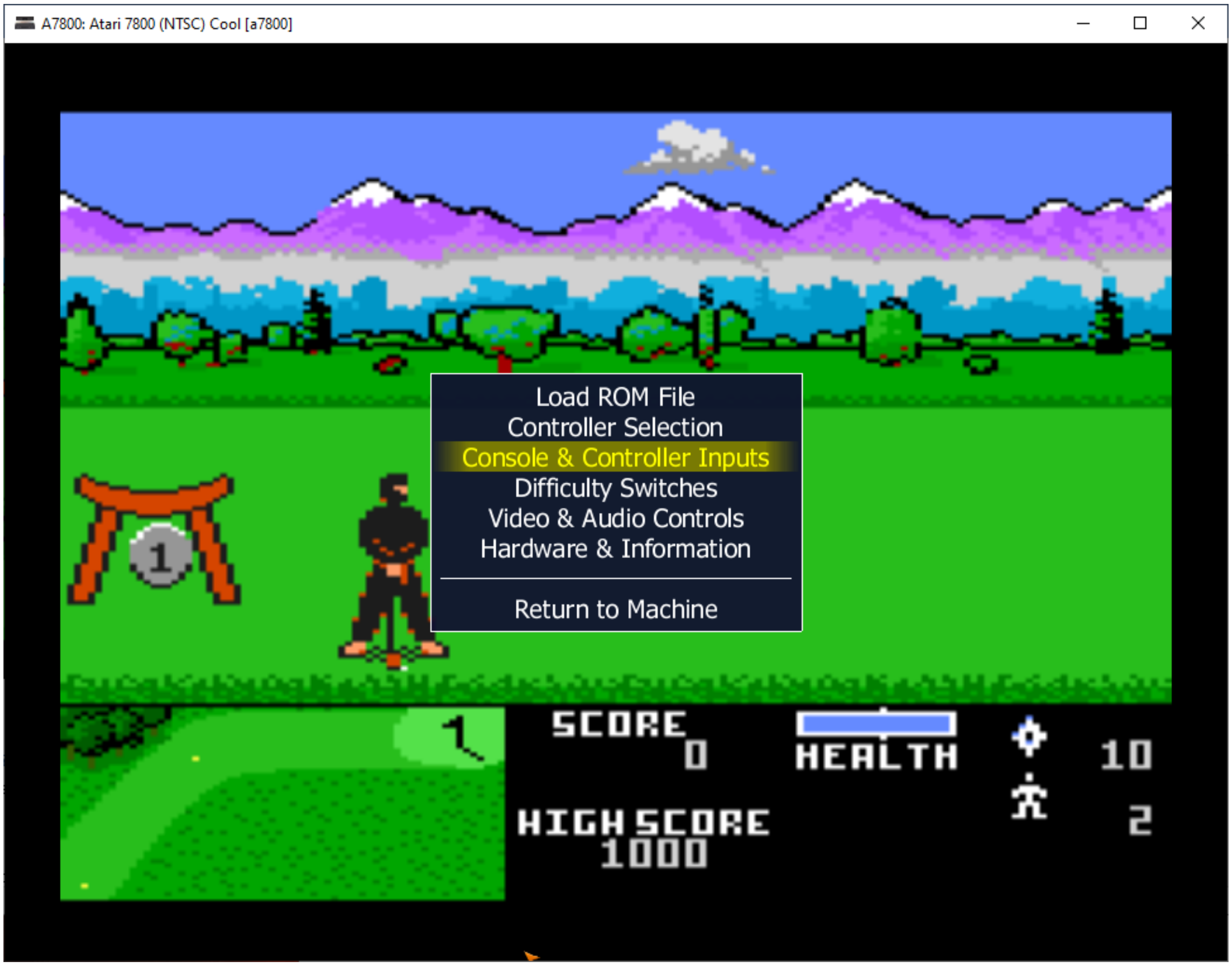
Controller Selection: View or change connected controller(s): *proline_joystick* (default), *vcs_joystick*, *lightgun*, *st_mouse*, *amiga mouse*, *c22_trakball*, *driving_wheel*, *paddle*, *keypad*, are all available.

Change a controller by highlighting it and then press either the left or right arrow key:



joy1 ◀lightgun▶
joy2 proline_joystick
cartslot a78_sg_pokey [internal]

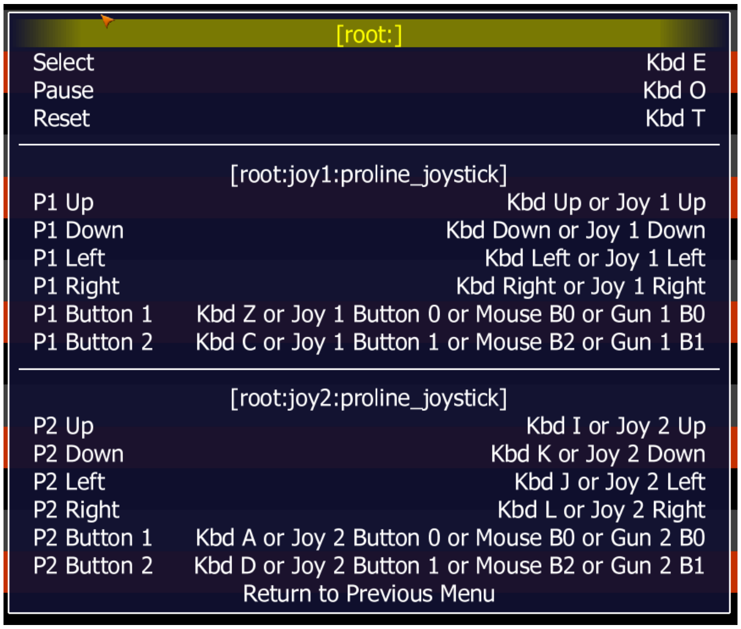
*<u>NOTE</u>: If a controller is changed, the emulator *must* be reset by simply double clicking with the mouse pointer, or navigating with the up and down arrow keys and pressing ‘Enter’, on the option to “Reset”.*



Console & Controller Inputs: Allows remapping how the keyboard and/or other devices interface with the functions of the Atari 7800 system.

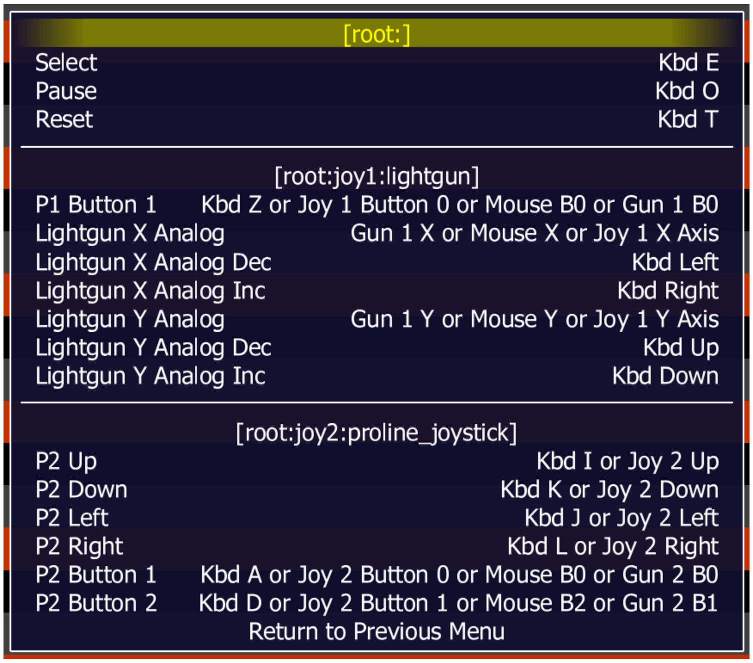
Maintained separately, each system has a set of default mappings as follows:

Console:	**Controller 1:**	**Controller 2:**
PAUSE = O	UP = Up Arrow	UP = I
SELECT = E	DOWN = Down Arrow	DOWN =K
RESET = T	LEFT = Left Arrow	LEFT = J
	RIGHT = Right Arrow	RIGHT = L
	BUTTON 1 = Z	BUTTON 1 = A
	BUTTON 2 = C	BUTTON 2 = D



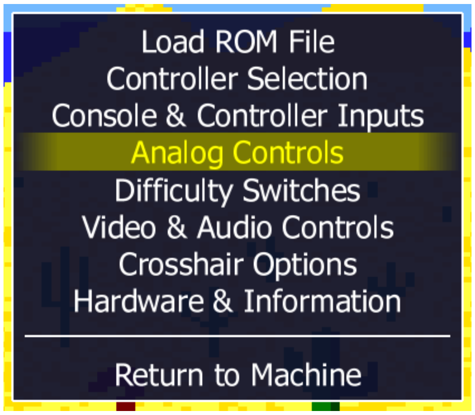
[root:]	
Select	Kbd E
Pause	Kbd O
Reset	Kbd T
[root:joy1:proline_joystick]	
P1 Up	Kbd Up or Joy 1 Up
P1 Down	Kbd Down or Joy 1 Down
P1 Left	Kbd Left or Joy 1 Left
P1 Right	Kbd Right or Joy 1 Right
P1 Button 1	Kbd Z or Joy 1 Button 0 or Mouse B0 or Gun 1 B0
P1 Button 2	Kbd C or Joy 1 Button 1 or Mouse B2 or Gun 1 B1
[root:joy2:proline_joystick]	
P2 Up	Kbd I or Joy 2 Up
P2 Down	Kbd K or Joy 2 Down
P2 Left	Kbd J or Joy 2 Left
P2 Right	Kbd L or Joy 2 Right
P2 Button 1	Kbd A or Joy 2 Button 0 or Mouse B0 or Gun 2 B0
P2 Button 2	Kbd D or Joy 2 Button 1 or Mouse B2 or Gun 2 B1
Return to Previous Menu	

The above window listing will change according to the controller device that is currently assigned. For instance, notice the change if a lightgun is mapped to Player 1 instead of a Proline joystick:

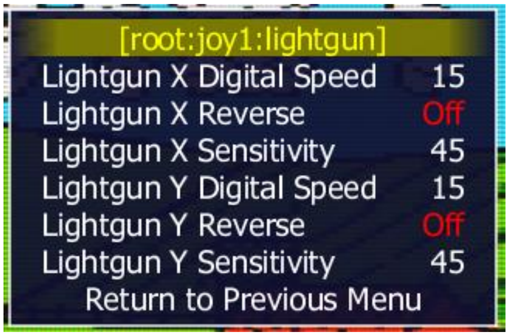


[root:]	
Select	Kbd E
Pause	Kbd O
Reset	Kbd T
[root:joy1:lightgun]	
P1 Button 1	Kbd Z or Joy 1 Button 0 or Mouse B0 or Gun 1 B0
Lightgun X Analog	Gun 1 X or Mouse X or Joy 1 X Axis
Lightgun X Analog Dec	Kbd Left
Lightgun X Analog Inc	Kbd Right
Lightgun Y Analog	Gun 1 Y or Mouse Y or Joy 1 Y Axis
Lightgun Y Analog Dec	Kbd Up
Lightgun Y Analog Inc	Kbd Down
[root:joy2:proline_joystick]	
P2 Up	Kbd I or Joy 2 Up
P2 Down	Kbd K or Joy 2 Down
P2 Left	Kbd J or Joy 2 Left
P2 Right	Kbd L or Joy 2 Right
P2 Button 1	Kbd A or Joy 2 Button 0 or Mouse B0 or Gun 2 B0
P2 Button 2	Kbd D or Joy 2 Button 1 or Mouse B2 or Gun 2 B1
Return to Previous Menu	

Additionally, after a controller such as the lightgun is chosen and the emulator is reset, new main menu items will appear such as “Analog Controls” or/and “Crosshair Options”:

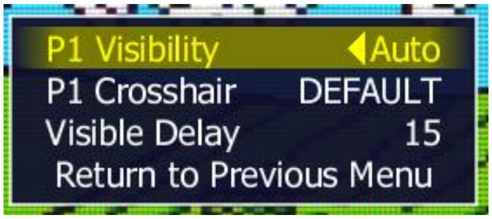


Analog Controls: Provisions speed, sensitivity, and reverse toggles in handling mouse movement and associated response:



```
[root:joy1:lightgun]
Lightgun X Digital Speed    15
Lightgun X Reverse         Off
Lightgun X Sensitivity      45
Lightgun Y Digital Speed    15
Lightgun Y Reverse         Off
Lightgun Y Sensitivity      45
   Return to Previous Menu
```

Crosshair Options: Controls how often and long crosshairs will be present on the screen:



```
P1 Visibility        ◀Auto
P1 Crosshair       DEFAULT
Visible Delay           15
 Return to Previous Menu
```

Other mappings - These cannot be modified via the available menus:

- A7800 menus = Tab
- Toggle between fullscreen and window modes = Alt+Enter
- Pause emulator = P
- Frame-by-frame pause advance = Shift+P
- Previous menu or exit emulator = Esc
- Reset emulator = F3
- View color palette = F4
- Save state = F6
- Load state = F7
- Decrease frame skip = F8
- Increase frame skip = F9
- Toggle emulation speed throttling = F10
- Display emulation speed = F11
- Take a screenshot* = F12
- Fast forward emulation** = Insert
- Toggle isolation of highlighted menu option = ~ [Tilde]

* *Sends and image to an already present "snap" folder.*

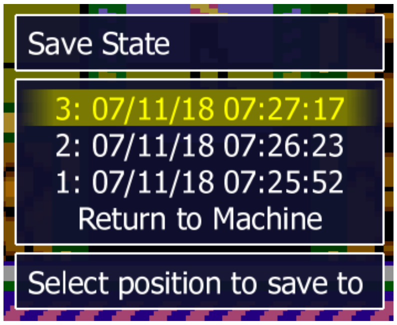
** *Runs a ROM without throttled speed and max frame skip when held down.*

Load & Save States

There are 36 real-time load and save states that can be utilized for each one of the six machines.

Pressing ‘F6’ will bring up dialogue boxes requesting a position to save the state. Valid choices are “0” thru “9” and “a” thru “z”. After a position is selected, the respective save state file is created with a corresponding date and time stamp attached to it.

Any already existing save states for the machine will be listed accordingly:

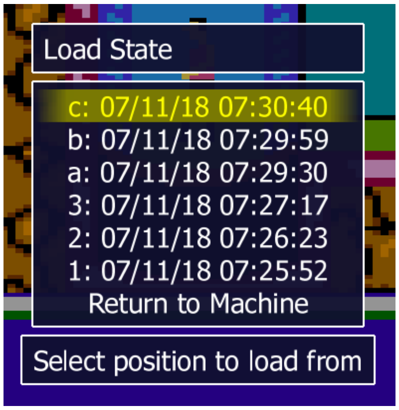


Save State

3: 07/11/18 07:27:17
2: 07/11/18 07:26:23
1: 07/11/18 07:25:52
Return to Machine

Select position to save to

Pressing ‘F7’ brings up dialogue boxes requesting a position to load from:

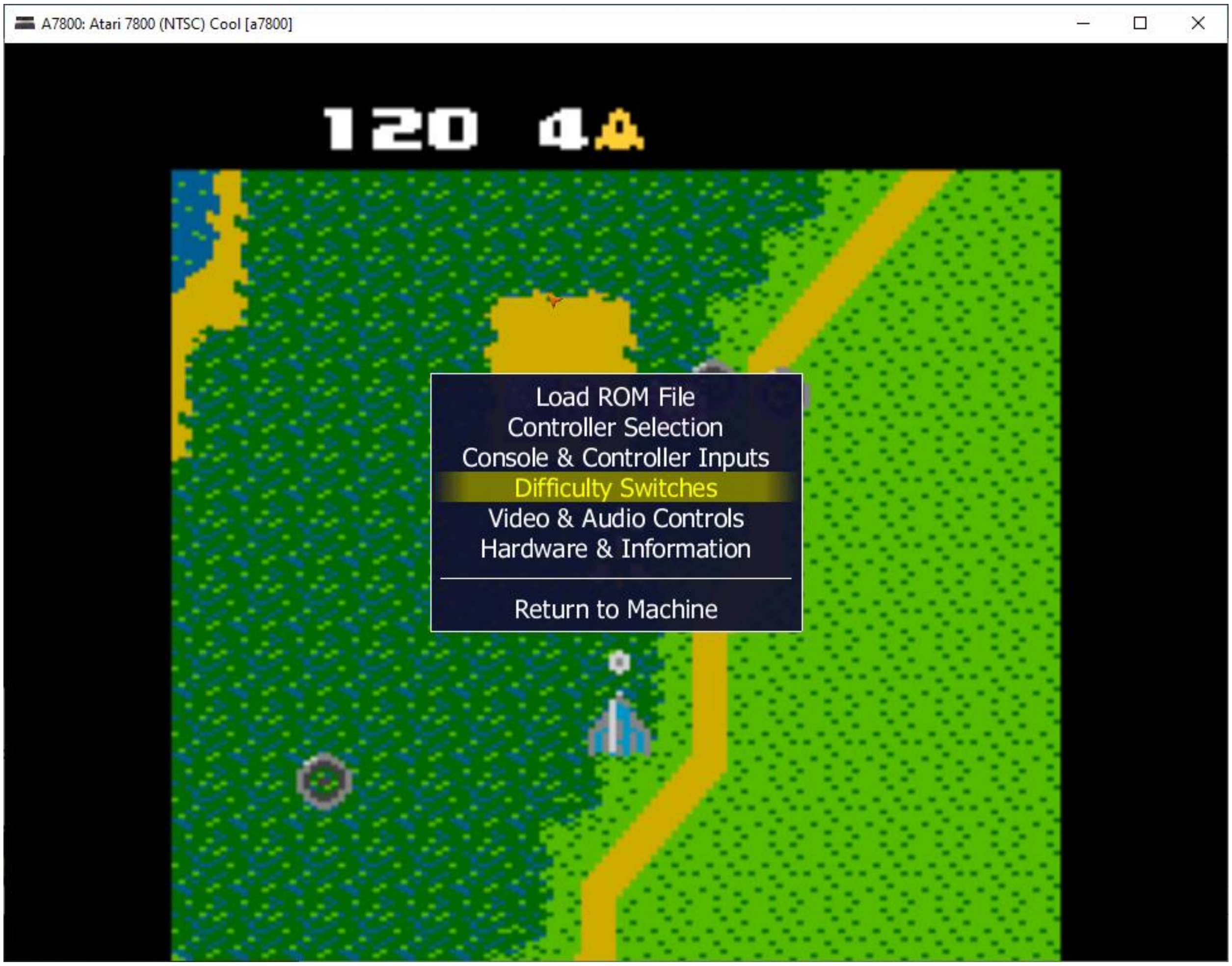


Load State

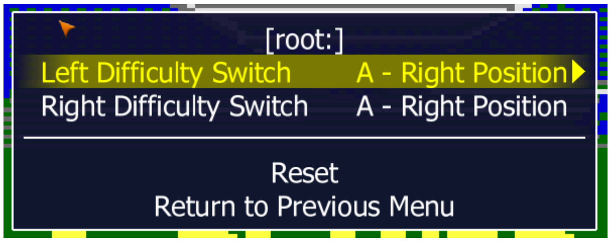
c: 07/11/18 07:30:40
b: 07/11/18 07:29:59
a: 07/11/18 07:29:30
3: 07/11/18 07:27:17
2: 07/11/18 07:26:23
1: 07/11/18 07:25:52
Return to Machine

Select position to load from

Ensure the correct ROM and any add-on hardware that was present during the save state creation, is in place prior to loading the state, in order for the function to perform successfully.



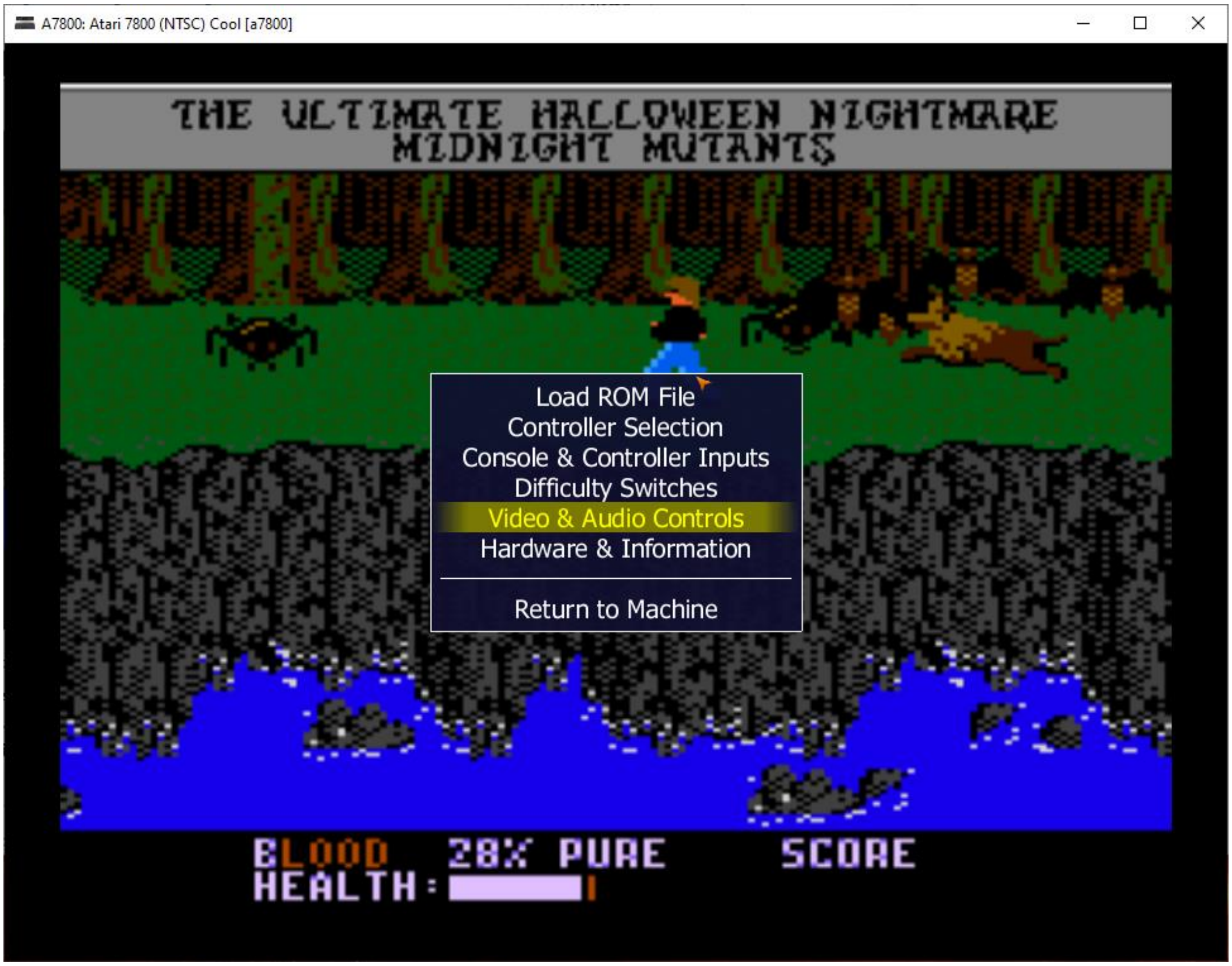
Difficulty Switches: Change the position of the difficulty switches found in front of the console.



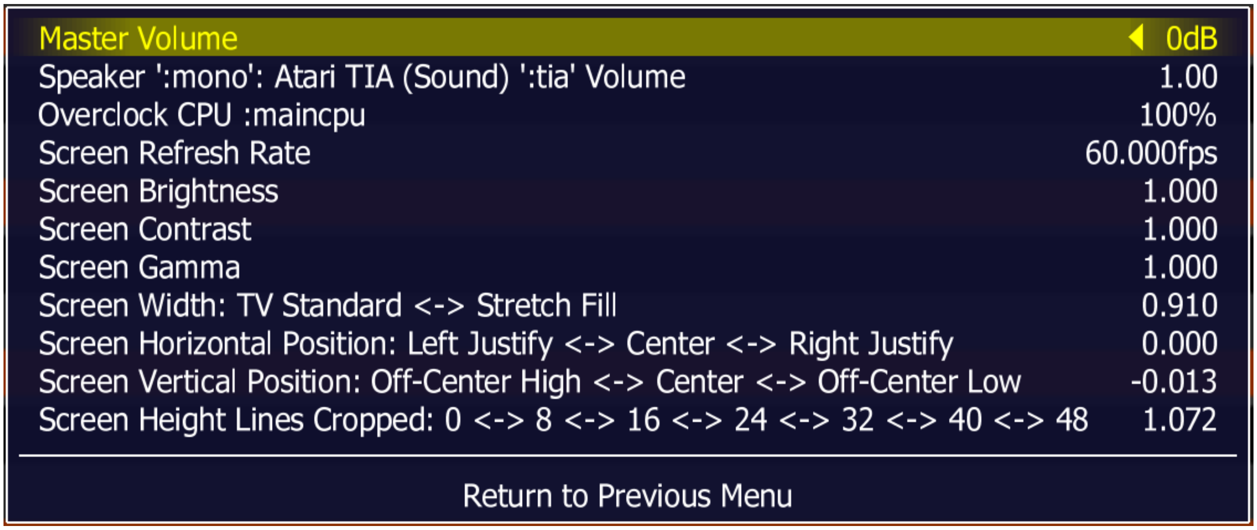
Left Difficulty Switch A - Right Position

Right Difficulty Switch A - Right Position

Each switch is changed by pressing the right or left arrow key. Whether or not a "Reset" is necessary falls strictly on the requirement of the game, dependent upon when and how difficulty switches are recognized. Some titles may require the switches to be set properly prior to start or power on, while others are impacted instantly.



Video and Audio Controls: Individual audio levels for TIA, POKEY, and other sound chips, when present. Brightness, contrast, gamma, as well as screen width, horizonal and vertical positioning, screen cropping video options are present too. Refresh Rate 'MARIA' adjustment performance and overclocking the 'SALLY' CPU are available here as well.

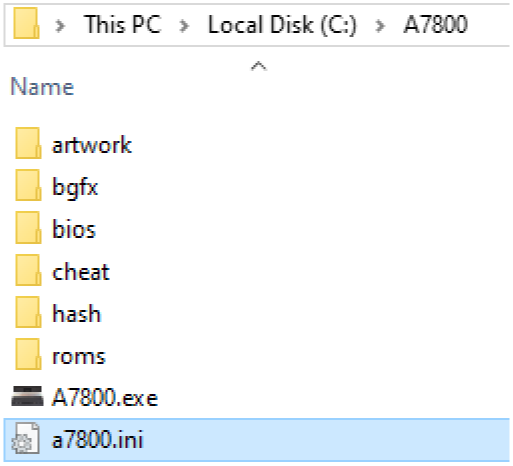


Master Volume	◀ 0dB
Speaker ':mono': Atari TIA (Sound) ':tia' Volume	1.00
Overclock CPU :maincpu	100%
Screen Refresh Rate	60.000fps
Screen Brightness	1.000
Screen Contrast	1.000
Screen Gamma	1.000
Screen Width: TV Standard <-> Stretch Fill	0.910
Screen Horizontal Position: Left Justify <-> Center <-> Right Justify	0.000
Screen Vertical Position: Off-Center High <-> Center <-> Off-Center Low	-0.013
Screen Height Lines Cropped: 0 <-> 8 <-> 16 <-> 24 <-> 32 <-> 40 <-> 48	1.072

Return to Previous Menu

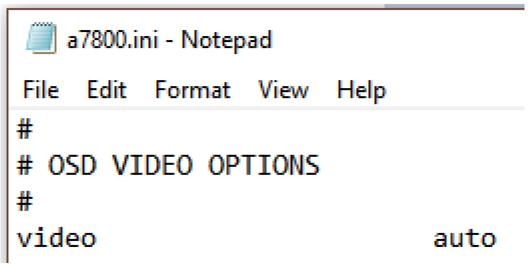
The majority of users should experience no issue with the default video of the emulator as “auto” finds and selects what is likely the best candidate according to OS. For Windows, it is "d3d". For OS X, it is "opengl". For all other platforms, it is "soft".

However, for any Windows users who experience issues or wishes for more out of their video experience, there are other video drivers that can be leveraged. By utilizing a simple text editor, open the "a7800.ini" file, located in the same folder as the A7800 executable; other OSes may find it under “;$HOME/.a7800/a7800.ini”:



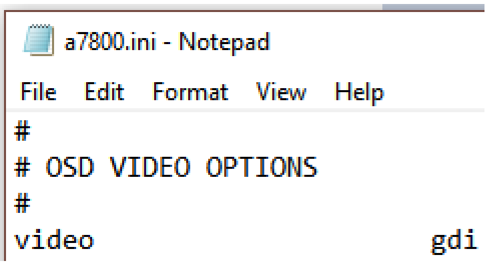
artwork
bgfx
bios
cheat
hash
roms
A7800.exe
a7800.ini

Change the "video" choice in the section noted below:



```
#
# OSD VIDEO OPTIONS
#
video                          auto
```

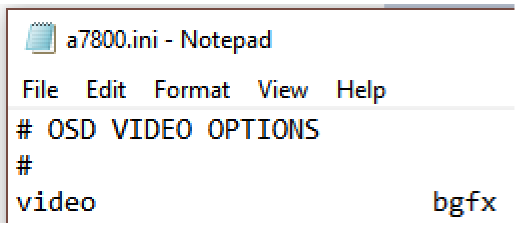
Two other video options include:



```
#
# OSD VIDEO OPTIONS
#
video                          gdi
```

Video GDI

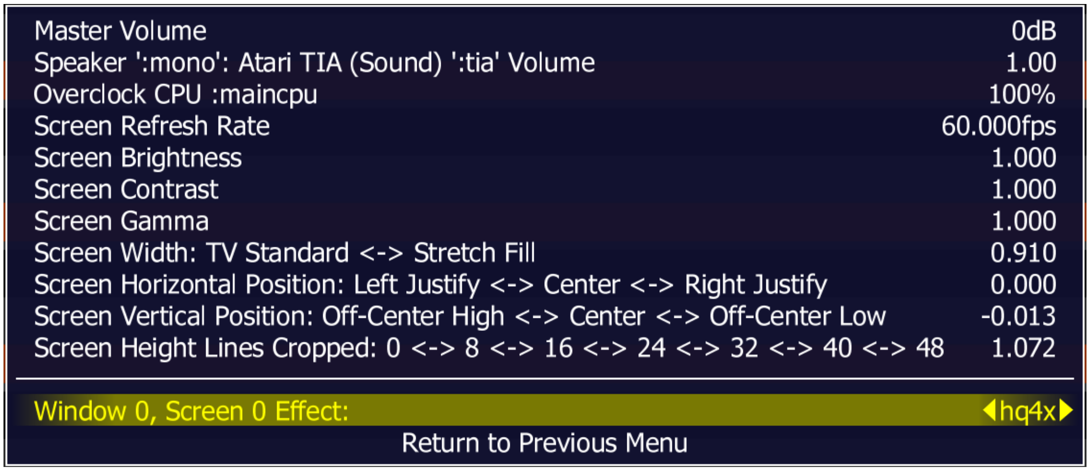
Utilizes an older Windows graphics standard, not the fastest but the most compatible choice for those with older graphics hardware or/and version of Windows.



```
# OSD VIDEO OPTIONS
#
video                          bgfx
```

Video BGFX

Hardware accelerated based renderer. This choice opens a plethora of video effects.



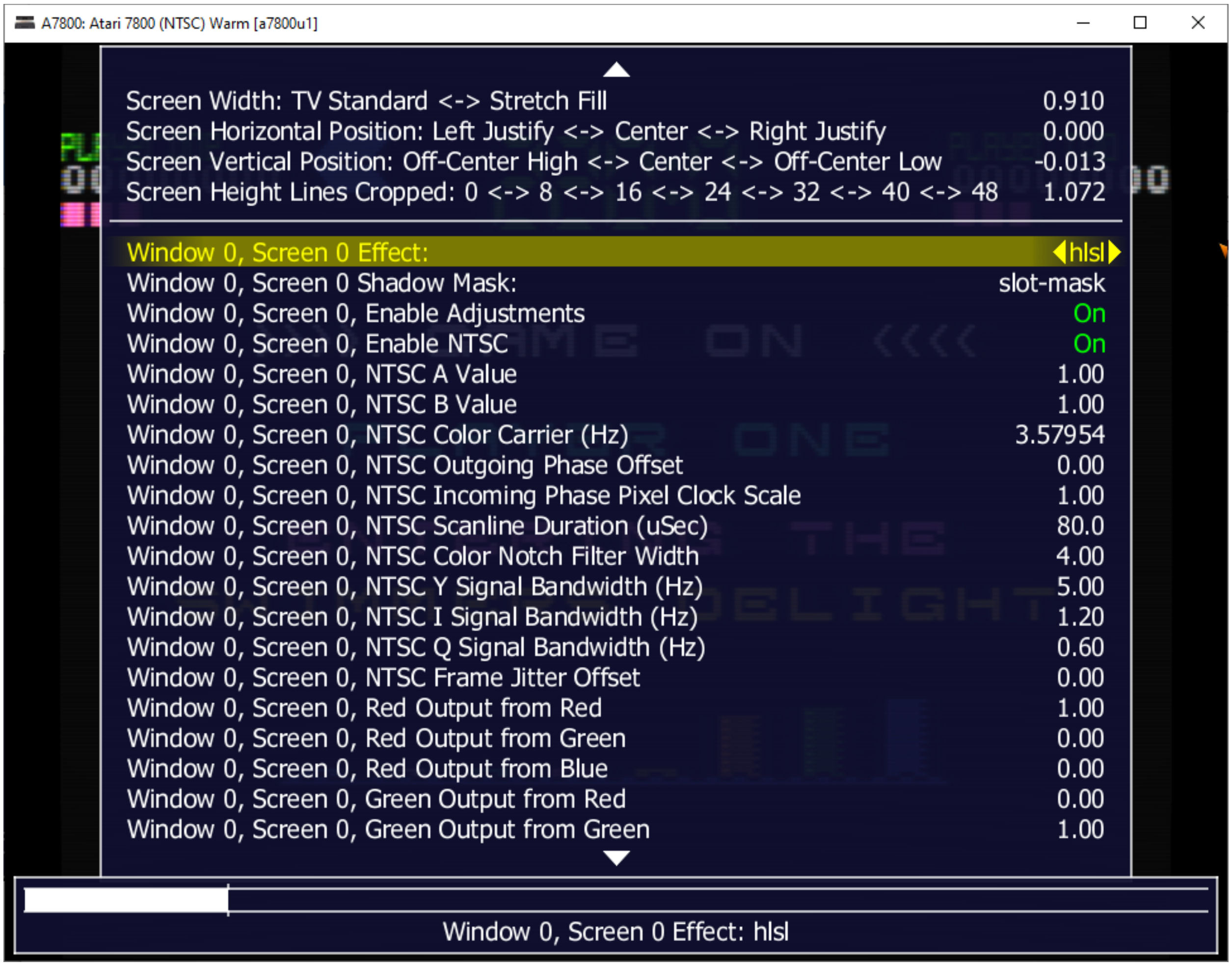
```
Master Volume                                                          0dB
Speaker ':mono': Atari TIA (Sound) ':tia' Volume                      1.00
Overclock CPU :maincpu                                                100%
Screen Refresh Rate                                              60.000fps
Screen Brightness                                                    1.000
Screen Contrast                                                      1.000
Screen Gamma                                                         1.000
Screen Width: TV Standard <-> Stretch Fill                           0.910
Screen Horizontal Position: Left Justify <-> Center <-> Right Justify 0.000
Screen Vertical Position: Off-Center High <-> Center <-> Off-Center Low  -0.013
Screen Height Lines Cropped: 0 <-> 8 <-> 16 <-> 24 <-> 32 <-> 40 <-> 48  1.072

Window 0, Screen 0 Effect:                                         ◀hq4x▶
                         Return to Previous Menu
```

To toggle through all the various video effects, simply highlight the "Window 0, Screen 0 Effect:" line, and press the left or right arrow key.

The complete list of base video effects is as follows:

- hlsl (includes optional NTSC/YIQ simulation),
- crt-geom-deluxe, crt-geom,
- super-eagle,
- hq2x, hq3x, hq4x,
- super-2xbr-3d-2p, super-2xbr-3d-3p-smoother, super-4xbr-3d-4p, super-4xbr-3d-6p-smoother, super-xbr-2p, super-xbr-3p-smoother, super-xbr-6p, super-xbr-deposterize, super-xbr-fast-3p, super-xbr-fast-6p, xbr-hybrid, xbr-lv1-noblend, xbr-lv2-3d, xbr-lv2-deposterize, xbr-lv2-fast, xbr-lv2-multipass, xbr-lv2-noblend, xbr-lv2, xbr-lv3-noblend, xbr-lv3.

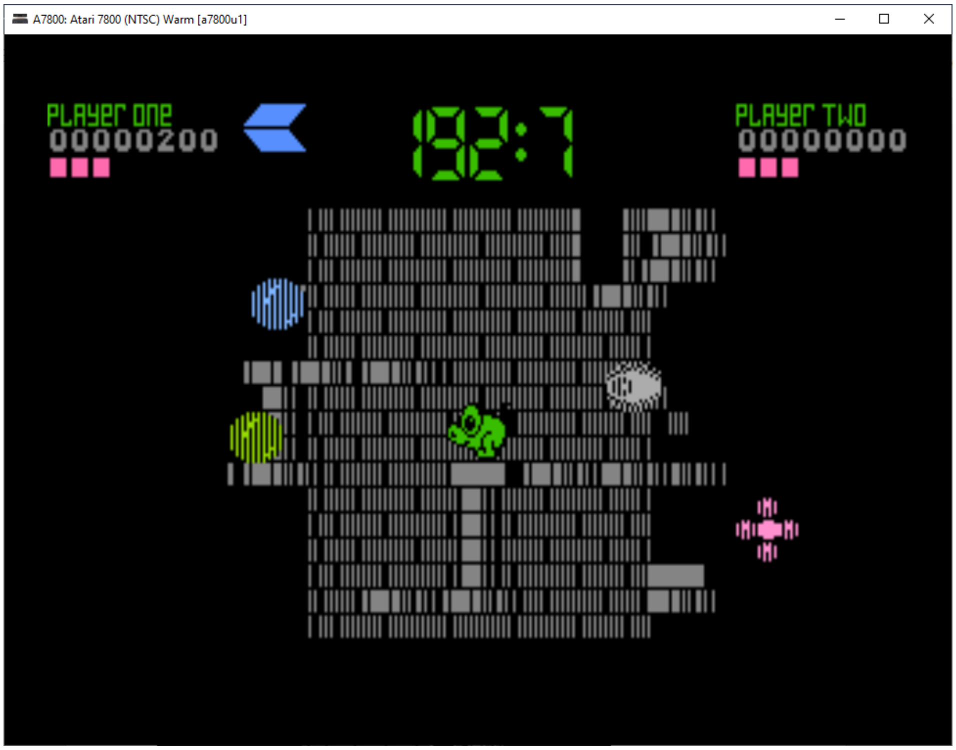
Within some of those effects are additional video options:

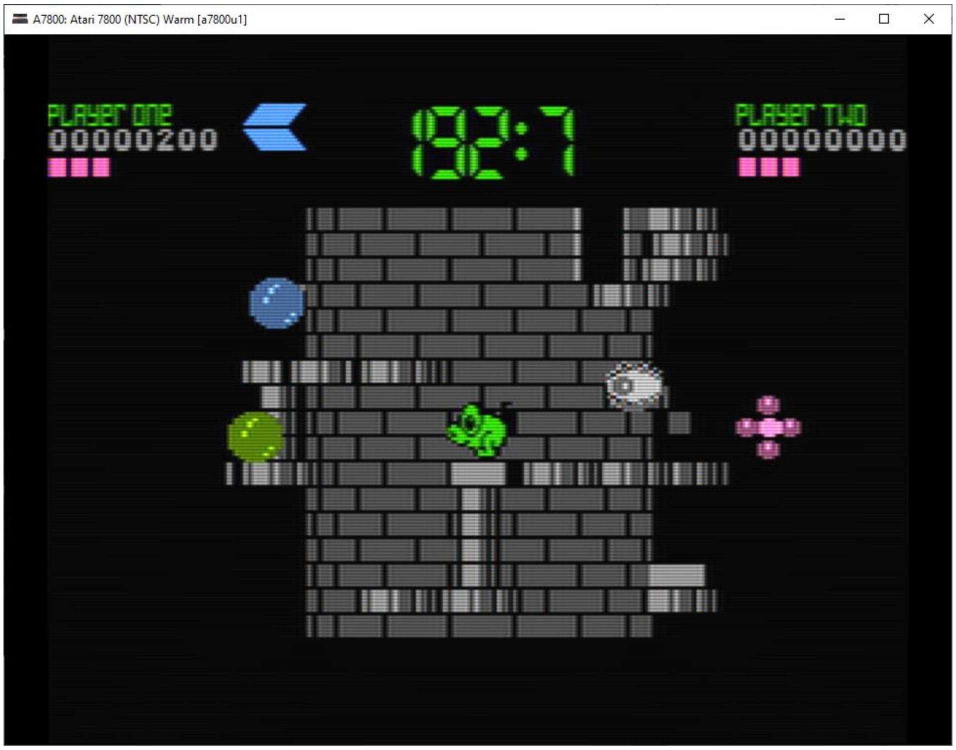


Screen Width: TV Standard <-> Stretch Fill	0.910
Screen Horizontal Position: Left Justify <-> Center <-> Right Justify	0.000
Screen Vertical Position: Off-Center High <-> Center <-> Off-Center Low	-0.013
Screen Height Lines Cropped: 0 <-> 8 <-> 16 <-> 24 <-> 32 <-> 40 <-> 48	1.072
Window 0, Screen 0 Effect:	◀hlsl▶
Window 0, Screen 0 Shadow Mask:	slot-mask
Window 0, Screen 0, Enable Adjustments	On
Window 0, Screen 0, Enable NTSC	On
Window 0, Screen 0, NTSC A Value	1.00
Window 0, Screen 0, NTSC B Value	1.00
Window 0, Screen 0, NTSC Color Carrier (Hz)	3.57954
Window 0, Screen 0, NTSC Outgoing Phase Offset	0.00
Window 0, Screen 0, NTSC Incoming Phase Pixel Clock Scale	1.00
Window 0, Screen 0, NTSC Scanline Duration (uSec)	80.0
Window 0, Screen 0, NTSC Color Notch Filter Width	4.00
Window 0, Screen 0, NTSC Y Signal Bandwidth (Hz)	5.00
Window 0, Screen 0, NTSC I Signal Bandwidth (Hz)	1.20
Window 0, Screen 0, NTSC Q Signal Bandwidth (Hz)	0.60
Window 0, Screen 0, NTSC Frame Jitter Offset	0.00
Window 0, Screen 0, Red Output from Red	1.00
Window 0, Screen 0, Red Output from Green	0.00
Window 0, Screen 0, Red Output from Blue	0.00
Window 0, Screen 0, Green Output from Red	0.00
Window 0, Screen 0, Green Output from Green	1.00

Window 0, Screen 0 Effect: hlsl

*<u>NOTE</u>: Some of the advanced video effects under bgfx are not for novice users or those uncomfortable with the notion of possibly spending a considerable amount of time tweaking. They are *advanced* video effects and incredible for at least a couple of reasons: 1. Resulting output. 2. The amount of time and effort it may take a user to get it 'right'.*





HLSL

For those looking to recreate the closest CRT experience possible from among the near 30 different video effects, it is highly recommended to utilize "hlsl". Inclusion of all bells and whistles of that effect requires setting "Enable NTSC" to “On”. It is turned on by default for A7800.

Of course, the more effects utilized the more hardware horsepower that is required. So, if some effects from hlsl are desired, but either the computer cannot handle the additional resource utilization or a cleaner look with some effects is the goal, turning “Enable NTSC” to “Off” would be best.

In short, it may be view as:

Enable NTSC = **On** → Composite output simulation.

Enable NTSC = **Off** → S-Video output simulation.

The following explanation of “hlsl” settings have been slightly edited from their source: http://docs.mamedev.org/advanced/hlsl.html

-shadow_mask_alpha (Shadow Mask Amount)

This defines how strong the effect of the shadowmask is. Acceptable range is from 0 to 1, where 0 will show no shadowmask effect, 1 will be a completely opaque shadowmask, and 0.5 will be 50% transparent.

-shadow_mask_tile_mode (Shadow Mask Tile Mode)

This defines whether the shadowmask should be tiled based on the screen resolution of your monitor or based on the source resolution of the emulated system. Valid values are 0 for Screen mode and 1 for Source mode.

-shadow_mask_texture
-shadow_mask_x_count (Shadow Mask Pixel X Count)
-shadow_mask_y_count (Shadow Mask Pixel Y Count)
-shadow_mask_usize (Shadow Mask U Size)
-shadow_mask_vsize (Shadow Mask V Size)
-shadow_mask_x_count (Shadow Mask U Offset)
-shadow_mask_y_count (Shadow Mask V Offset)

These settings need to be set in unison with one another. In particular, shadow_mask_texture sets rules for how you need to set the other options.

-shadow_mask_texture

Sets the texture of the shadowmask effect.

-shadow_mask_usize and shadow_mask_vsize

Define the used size of the shadow_mask_texture in percentage, staring at the top-left corner. The means for a texture with the actual size of 24x24 pixel and an u/v size of 0.5,0.5 the top-left 12x12 pixel will be used. Keep in mind to define an u/v size that makes is possible to tile the texture without gaps or glitches. 0.5,0.5 is fine for any shadowmask texture that is included with MAME.

-shadow_mask_x_count and shadow_mask_y_count

Define how many screen pixel should be used to display the u/v sized texture. e.g. if you use the example from above and define a x/y count of 12,12 every pixel of the texture will be displayed 1:1 on the screen, if you define a x/y count of 24,24 the texture will be displayed twice as large.

Examples…

For shadow_mask.png:

```
shadow_mask_texture shadow-mask.png
shadow_mask_x_count 12
shadow_mask_y_count 6 or 12
shadow_mask_usize 0.5
shadow_mask_vsize 0.5
```

For slot-mask.png:

```
shadow_mask_texture slot-mask.png
shadow_mask_x_count 12
shadow_mask_y_count 8 or 16
shadow_mask_usize 0.5
shadow_mask_vsize 0.5
```

For aperture-grille:

```
shadow_mask_texture aperture-grille.png
shadow_mask_x_count 12
shadow_mask_y_count 12 or any
shadow_mask_usize 0.5
shadow_mask_vsize 0.5
```

-shadow_mask_uoffset and shadow_mask_voffset

Can be used to tweak the alignment of the final shadowmask in subpixel range. Range is from -1.00 to 1.00, where 0.5 moves the shadowmask by 50 percent of the u/v sized texture.

-distortion (Quadric Distortion Amount)

This setting determines strength of the quadric distortion of the screen image.

-cubic_distortion (Cubic Distortion Amount)

This setting determines strength of the qubic distortion of the screen image.

<u>Note</u>: Both distortion factors can be negative to compensate each other. e.g. distortion 0.5 and cubic_distortion -0.5

-distort_corner (Distorted Corner Amount)

This setting determines strength of distortion of the screen corners, which does not affect the distortion of screen image itself.

-round_corner (Rounded Corner Amount)

The corners of the display can be rounded off through the use of this setting.

-smooth_border (Smooth Border Amount)

Sets a smoothened/blurred border around the edges of the screen.

-reflection (Reflection Amount)

If set above 0, this creates a white reflective blotch on the display. By default, this is put in the upper right corner of the display. By editing the POST.FX file's GetSpotAddend section, you can change the location. Range is from 0.00 to 1.00.

-vignetting (Vignetting Amount)

When set above 0, will increasingly darken the outer edges of the display in a pseudo-3D effect. Range is from 0.00 to 1.00.

-scanline_alpha (Scanline Amount)

This defines how strong the effect of the scanlines are. Acceptable range is from 0 to 1, where 0 will show no scanline effect, 1 will be a completely black line, and 0.5 will be 50% transparent. Note that arcade monitors did not have completely black scanlines.

-scanline_size (Overall Scanline Scale)

The overall spacing of the scanlines is set with this option. Setting it at 1 represents consistent alternating spacing between display lines and scanlines.

-scanline_height (Individual Scanline Scale)

This determines the overall size of each scanline. Setting lower than 1 makes them thinner, larger than 1 makes them thicker.

-scanline_variation (Scanline Variation)

This affects the size of each scanline depending on its brightness. Brighter scanlines will be thicker than darker scanline. Acceptable range is from 0 to 2.0, with the default being 1.0. At 0.0 all scanlines will have the same size independent of their brightness.

-scanline_bright_scale (Scanline Brightness Scale)

Specifies how bright the scanlines are. Larger than 1 will make them brighter, lower will make them dimmer. Setting to 0 will make scanlines disappear entirely.

-scanline_bright_offset (Scanline Brightness Offset)

This will give scanlines a glow/overdrive effect, softening and smoothing the top and bottom of each scanline.

-scanline_jitter (Scanline Jitter Amount)

Specifies the wobble or jitter of the scanlines, causing them to jitter on the monitor. Warning: Higher settings may hurt your eyes.

-hum_bar_alpha (Hum Bar Amount)

Defines the strength of the hum bar effect.

-defocus (Defocus)

This option will defocus the display, blurring individual pixels like an extremely badly maintained monitor. Specify as X,Y values (e.g. defocus 1,1)

-converge_x (Linear Convergence X, RGB)
-converge_y (Linear Convergence Y, RGB)
-radial_converge_x (Radial Convergence X, RGB)
-radial_converge_y (Radial Convergence Y, RGB)

Adjust the convergence of the red, green, and blue channels in a given direction. Many badly maintained monitors with bad convergence would bleed colored ghosting off-center of a sprite, and this simulates that.

-red_ratio (Red Output from RGB)
-grn_ratio (Green Output from RGB)
-blu_ratio (Blue Output from RGB)

Defines a 3x3 matrix that is multiplied with the RGB signals to simulate color channel interference. For instance, a green channel of (0.100, 1.000, 0.250) is weakened 10% by the red channel and strengthened 25% through the blue channel.

-offset (Signal Offset)

Strengthen or weakens the current color value of a given channel. For instance, a red signal of 0.5 with an offset of 0.2 will be raised to 0.7

-scale (Signal Scale)

Applies scaling to the current color value of the channel. For instance, a red signal of 0.5 with a scale of 1.1 will result in a red signal of 0.55

-power (Signal Exponent, RGB)

Exponentiate the current color value of the channel, also called gamma. For instance, a red signal of 0.5 with red power of 2 will result in a red signal of 0.25

-floor (Signal Floor, RGB)

Sets the absolute minimum color value of a channel. For instance, a red signal of 0.0 (total absence of red) with a red floor of 0.2 will result in a red signal of 0.2

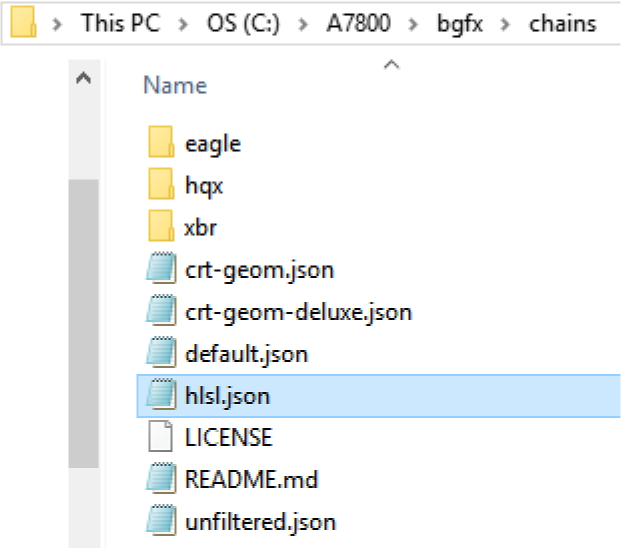
-phosphor_life (Phosphor Persistence, RGB)

How long the color channel stays on the screen, also called phosphor ghosting. 0 gives absolutely no ghost effect, and 1 will leave a contrail behind that is only overwritten by a higher color value.

-saturation (Color Saturation)

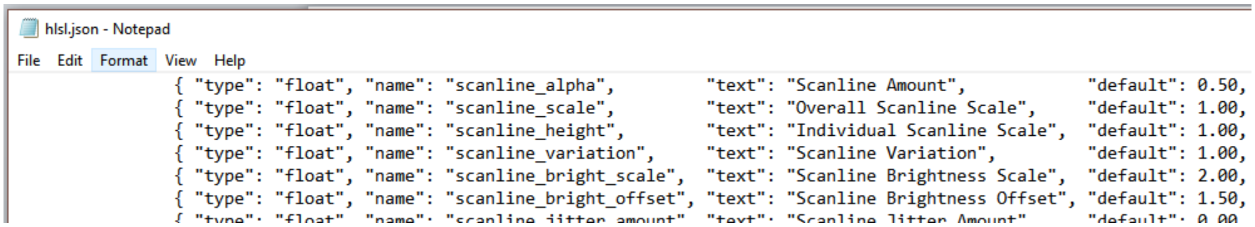
Color saturation can be adjusted here.

All bgfx effects will reset back to default upon changing from full screen to window, or when exiting/restarting the emulator. Settings do not save automatically. In order to retain the desired effects, the file “hlsl.json” needs to be edited and saved accordingly.

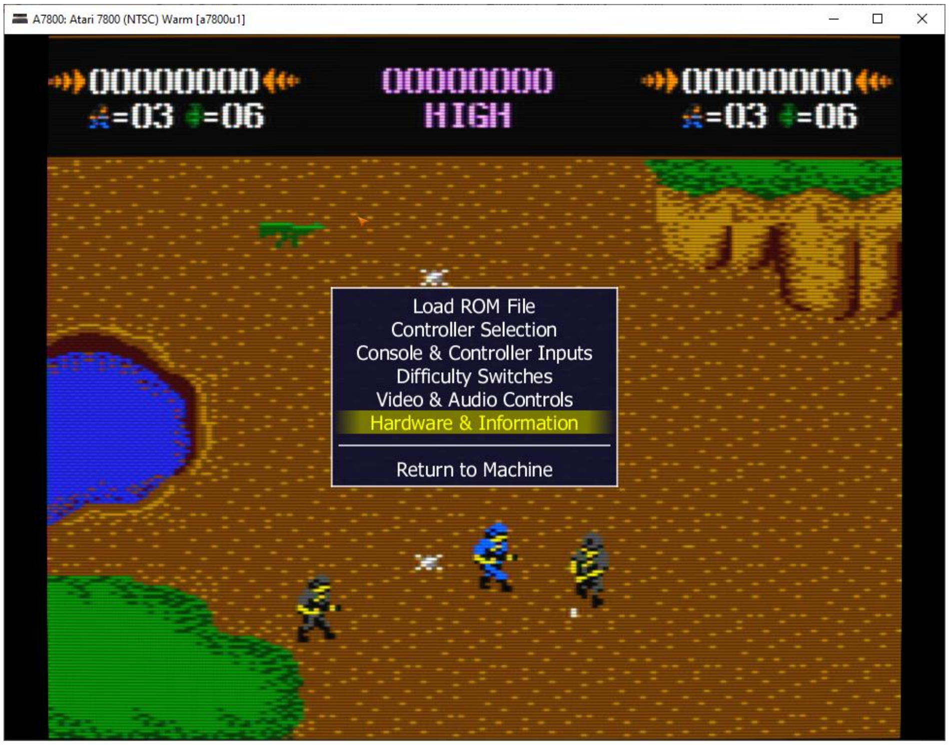

This PC > OS (C:) > A7800 > bgfx > chains

eagle
hqx
xbr
crt-geom.json
crt-geom-deluxe.json
default.json
hlsl.json
LICENSE
README.md
unfiltered.json

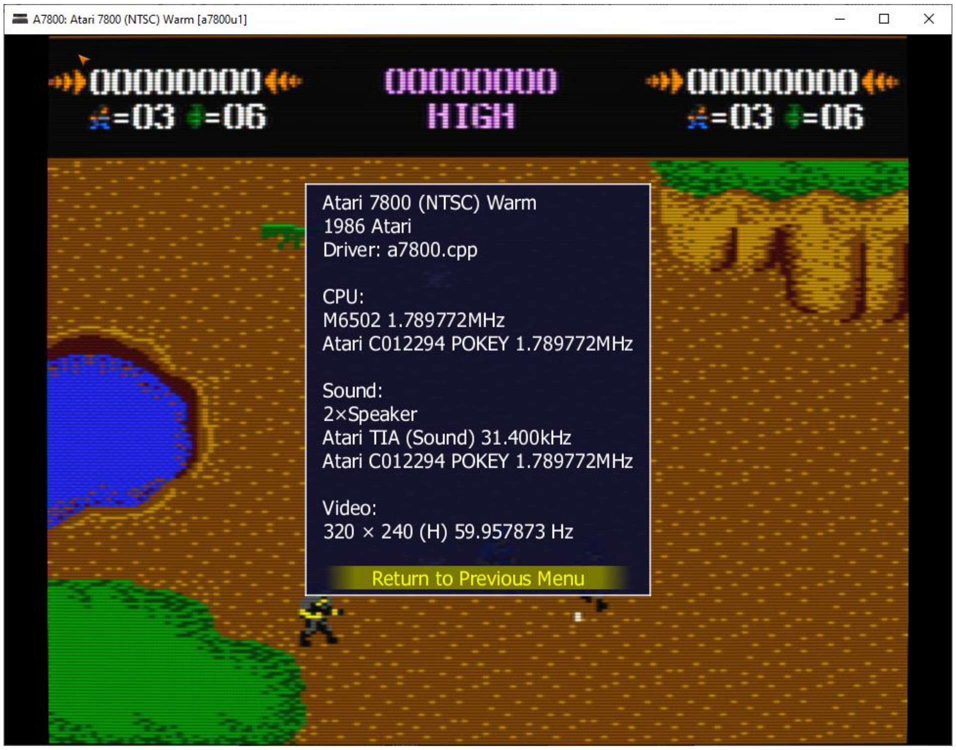
As is the case with a7800.ini, hlsl.json can be opened utilizing a simple text editor (I.E. Notepad), modified, and then saved:



```
                { "type": "float", "name": "scanline_alpha",         "text": "Scanline Amount",          "default": 0.50,
                { "type": "float", "name": "scanline_scale",         "text": "Overall Scanline Scale",   "default": 1.00,
                { "type": "float", "name": "scanline_height",        "text": "Individual Scanline Scale", "default": 1.00,
                { "type": "float", "name": "scanline_variation",     "text": "Scanline Variation",       "default": 1.00,
                { "type": "float", "name": "scanline_bright_scale",  "text": "Scanline Brightness Scale", "default": 2.00,
                { "type": "float", "name": "scanline_bright_offset", "text": "Scanline Brightness Offset", "default": 1.50,
                { "type": "float", "name": "scanline_jitter_amount", "text": "Scanline Jitter Amount",   "default": 0.00
```



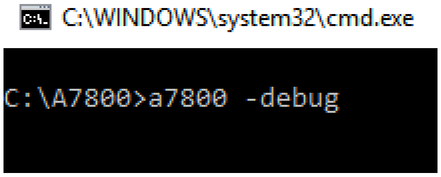
Hardware & Information: Provides specifics on the console and options selected including release year, the driver name, CPU, sound chip(s), and video resolution. Some section information will change depending on the hardware present (I.E. XM Expansion Module, sound chip on cart, etc.)



```
Atari 7800 (NTSC) Warm
1986 Atari
Driver: a7800.cpp

CPU:
M6502 1.789772MHz
Atari C012294 POKEY 1.789772MHz

Sound:
2×Speaker
Atari TIA (Sound) 31.400kHz
Atari C012294 POKEY 1.789772MHz

Video:
320 × 240 (H) 59.957873 Hz
```

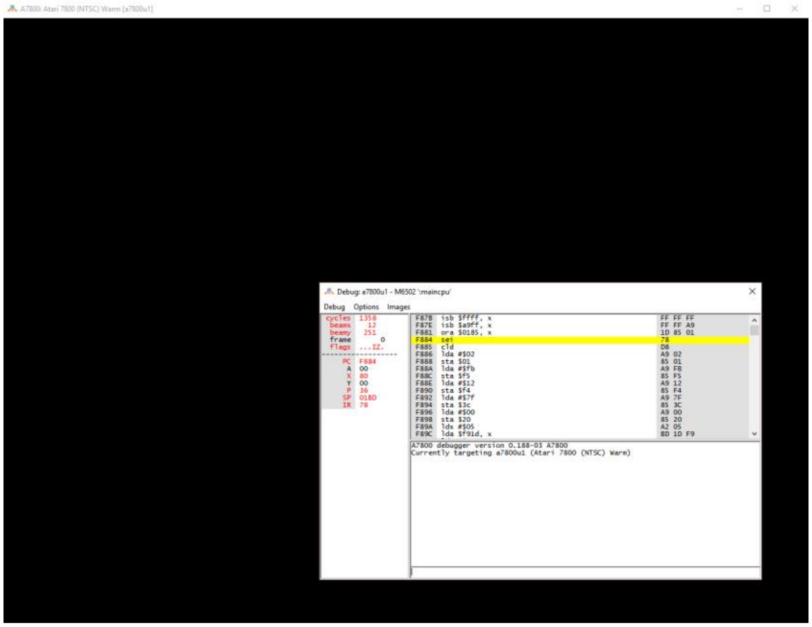
Debugger

A7800 has a built-in debugger inherited from MAME that is made active by starting the emulator with the “-debug” switch from the Command Line:



```
C:\A7800>a7800 -debug
```

Once it is launched, proceed with selecting a console region-temperature, and then system configuration, as explained earlier in this manual. After doing so, two windows will appear; the typical emulation window display and a smaller debugger window:


A7800: Atari 7800 (NTSC) Warm [a7800u1]
Debug: a7800u1 - M6502 ':maincpu'
Debug
Options
Images

```
cycles  1358
 beamx    12
 beamy   251
 frame      0
 flags  ....IZ.
-----------------
    PC  F884
     A  00
     X  80
     Y  00
     P  36
    SP  01BD
    IR  78
```

```
F87B  isb $ffff, x     FF FF FF
F87E  isb $a9ff, x     FF FF A9
F881  ora $0185, x     1D 85 01
F884  sei              78
F885  cld              D8
F886  lda #$02         A9 02
F888  sta $01          85 01
F88A  lda #$fb         A9 FB
F88C  sta $f5          85 F5
F88E  lda #$12         A9 12
F890  sta $f4          85 F4
F892  lda #$7f         A9 7F
F894  sta $3c          85 3C
F896  lda #$00         A9 00
F898  sta $20          85 20
F89A  ldx #$05         A2 05
F89C  lda $f91d, x     BD 1D F9
```

```
A7800 debugger version 0.188-03 A7800
Currently targeting a7800u1 (Atari 7800 (NTSC) Warm)
```

Press ‘F5’ to toggle the debugger window from being active and allow the main emulation window to become the active one. Press the ‘Tab’ key to bring up the emulator’s main menu and load the desired ROM.

After a ROM is loaded the debugger is the active window again and the ROM program is paused. At this point the debugger is awaiting a command to execute.

List of general debugger commands:

do: evaluates the given expression.

symlist: lists registered symbols.

softreset: executes a soft reset.

hardreset: executes a hard reset.

print: prints one or more <item>s to the console.

printf: prints one or more <item>s to the console using <format>.

logerror: outputs one or more <item>s to the error.log.

tracelog: outputs one or more <item>s to the trace file using <format>.

tracesym: outputs one or more <item>s to the trace file.

history: outputs a brief history of visited opcodes.

trackpc: visually track visited opcodes [boolean to turn on and off, for the given cpu, clear].

trackmem: record which PC writes to each memory address [boolean to turn on and off, clear].

pcatmem: query which PC wrote to a given memory address for the current CPU.

rewind: go back in time by loading the most recent rewind state.

statesave: save a state file for the current machine and ROM configuration.

stateload: load a state file for the current machine and ROM configuration.

snap: save a screen snapshot.

source: reads commands from <filename> and executes them one by one.

quit: exits A7800 and the debugger.

Typing "**help <command>**" in the A7800 debugger interface will provide further details on each command except "history".

FAQs

Q: *What do I need to run this emulator?*

A: At a minimum you'll need to obtain an Atari 7800 BIOS image. You may also wish to obtain the High Score Cart BIOS image as well. ROM and BIOS images aren't provided with A7800.

Q: *Why not just contribute this work to the MAME team, so I don't have to download this thing?*

A: This work was initially submitted to MAME team and rejected. MAME devs wish to have Proline 2-button controllers (and any other new DB-9 controllers) assigned directly to the VCS. Since this would involve integration, testing, and changes to the VCS driver and other drivers that may use it, the extra work involved wasn't worth it to us. In the end, MAME devs have broader wishes, and 7800.8bitdev.org is focused on the 7800.

Q: *Do you have any tips or guides to assist further with the debugger?*

A: See the A7800.8bitdev.org page regarding the MAME debugger:
http://7800.8bitdev.org/index.php/Introduction_to_the_MAME_debugger

For an extensive command list with detailed explanations, see the docs.mamedev.org pages:
http://docs.mamedev.org/debugger/index.html