Snes9x: The Portable Super Nintendo Entertainment System Emulator

v1.42 4-Dec-2003

Home page: http://www.snes9x.com

Contents

Changes Since Last Release

Check the CHANGES file for a complete history of Snes9x changes between versions.

Introduction

Snes9x is a portable, freeware Super Nintendo Entertainment System (SNES) emulator. It basically allows you to play most games designed for the SNES and Super Famicom Nintendo game systems on your PC or Workstation. The games include some real gems that were only ever released in Japan.

Snes9x is the result of well over six years worth of part-time hacking, coding, recoding, debugging, etc. Snes9x is coded in C++, with three assembler CPU emulation cores on the i386 Linux, DOS and Windows ports.

Snes9x is better than a real SNES:

- Freeze a game at any position, then restore the game to that exact spot at a later date - ideal for saving a game just before a difficult bit.
- Built-in cheat cartridge.
- Built-in peripheral emulation. The SNES mouse, Multi-player 5 and SuperScope external add-ons are all emulated, they cost extra money with a real SNES.
- Stereo sound - yes I know the SNES produced stereo sound, but who actually paid the inflated price for the special lead just so you could hear it?

o No more cartridge contact cleaning!
o Some SNES hardware features that be turned on and off during game play, games might be using one of these features to deliberately make a section of the game more difficult. Easy, just turn the feature off!
o Networked game play on some ports.
o Speed up or slow down SNES games.
o Save screen shots to impress(?!) your friends. ('F12')
o No unsightly stacks of cartridges.

Snes9x is worse than a real SNES:
o Unless your computer is very fast (Pentium II+), some games just can't hit every frame being rendered and the emulator starts to skip the drawing of some frames to keep the emulator running at a constant speed - to you it appears as if the graphics aren't moving as smoothly as they could.
o Not all games work; bugs and missing features cause some games to fail to work or renders them un-playable.
o You have to wait for your computer to boot before you can play games, no waiting on the real SNES!
o The SNES has an analogue low-pass sound filter that give a nice bass to all the sounds and music - Snes9x doesn't emulate this. If you have a posh sound card, you could try fiddling with it mixer controls to produce a similar effect. Turning on interpolated sound helps a lot.

What's Emulated

- The 65c816 main CPU.
- The Sony SPC700 sound CPU.
- SNES variable length machine cycles.
- 8 channel DMA and H-DMA (raster effects).
- All background modes, 0 to 7.
- Sound DSP, with eight 16-bit, stereo channels, compressed samples, hardware attack-decay-sustain-release volume processing, echo, pitch modulation and digital FIR sound filter.
- 8x8, 16x8 and 16x16 tile sizes, flipped in either direction.
- 32x32, 32x64, 64x32 and 64x64 screen tile sizes.
- H-IRQ, V-IRQ and NMI.
- Mode 7 screen rotation, scaling and screen flipping.
- Vertical offset-per-tile in modes 2, and 4.
- Horizontal offset-per-tile in modes 2, 4 and 6.
- 256x224, 256x239, 512x224, 512x239, 512x448 and 512x478 SNES screen resolutions.
- Sub-screen and fixed colour blending effects.
- Mosaic effect.
- Single and dual graphic clip windows, with all four logic combination modes.
- Colour blending effects only inside or outside a window.
- 128 8x8, 16x16, 32x32 or 64x64 sprites, flipped in either direction.
- SNES palette changes during frame (15/16-bit internal rendering only).
- Direct colour mode - uses tile and palette-group data directly as RGB value.
- Super FX, a 21/10MHz RISC CPU found in the cartridge of several games.
- S-DD1, a data decompression chip used only in Star Ocean and Street Fighter 2 Alpha. The compression algorithm is integrated into Snes9x, but you may still use the old graphics pack cheat as a speed boost.
- SPC7110, similar in use to S-DD1, but the algorithm is still unknown.
- S-RTC, a real-time clock chip. Dai Kaijyu Monogatari II is the only game that uses it.
- SA-1, a faster version of CPU found in the main SNES unit together with some custom game-accelerator hardware.
- C4, a custom Capcom chip used only in Megaman X2 and X3. Its a sprite scaler/rotator/line drawer/simple maths co-processor chip used to enhance some in-game effects.

- OBC1 is a sprite management chip. Metal combat is the only game to use this.
- Greater DSP-1 support, enough that all games should load, but some may have graphical glitches.
- DSP-2 support. Only used in Dungeon Master
- SNES mouse.
- SuperScope (light gun) emulated using computer mouse.
- Multi-player 5 - allowing up to five people to play games simultaneously on games that support that many players.
- Game-Genie and Action Replay cheat codes.
- Multiple ROM image formats, with or without a 512 byte copier header.
- Single or split images, compressed using zip and gzip, and interleaved in one of two ways.
- Auto S-RAM (battery backed RAM) loading and saving.
- Freeze-game support, now portable between different Snes9x ports.
- Interpolated sound.

What's Not
==========
- Only partial DSP-1 support, enough to play Mario Kart, Pilotwings and many others. All DSP-1 games should boot, but may display graphical errors.
- Any other odd chips that manufactures sometimes placed inside the cartridge to enhance games and as a nice side-effect, also act as an anti-piracy measure. (Examples: DSP-X, Seta, etc)
- Pseudo hi-res. mode - SNES hardware uses interpolation to give apparent increase in horizontal resolution, which is only partially emulated.
- Mosaic effect on mode 7.
- A couple of SPC700 instructions that I can't work exactly out what they should do.
- Fixed colour and mosaic effects in SNES hi-res. (512x448) modes.
- Offset-per-tile in mode 6. Luckily I haven't found a game that uses it, yet.
- The expansion slot found in many carts.

What You Will Need
==================

Windows 95, 98, NT, ME, 2K, or XP with at least DirectX 6 and OpenGL installed running on a modern, fast (e.g. Pentium 200 or higher) computer with at least 32Mb of RAM. Some games require another CPU to be emulated and/or make heavy use of colour translucency effects, so an even faster computer may be required to get an acceptable frame rate.

If you're running the original version of Windows 95 you will need to download Microsoft's OpenGL upgrade kit from there web site; they keep moving the exact location, but last time I looked I found it by using the following method: http://www.microsoft.com -> support -> Product support options -> download centre -> keyword search: OpenGL, product: Windows 95; the OpenGL DLL install package was the first option in the list after pressing Find It!

If you want SNES sound emulation, you'll need a DirectSound compatible sound card - virtually all modern PCI sound cards are DirectSound compatible - or use FMOD's older Windows WAVE sound driver.

Snes9x's full-screen mode uses DirectDraw to switch to the required resolution and depth, but if you intend to use the Windowed mode, for maximum emulation speed you should have your desktop depth set to 256 colours if translucency emulation and 16-bit rendering are switched off and not required, or hi-colour mode (32768/65536 colours) if translucency effects are required.

If you have a Voodoo 3dfx card, Snes9x can use this hardware to stretch and

smooth the relatively lo-resolution SNES image to fill your computer screen. Newer nvidia cards can do the same trick, just select stretch image option. There is also an OpenGL display output mode that can stretch and smooth the SNES image. If your OpenGL hardware supports 16-bit textures (most do), OpenGL mode is as fast, or faster than Voodoo 3dfx mode.

Software

Access to SNES ROM images in *.smc, *.sfc, *.fig or *.1, *.2, or sf32xxxa, sf32xxxb, etc., format otherwise you will have nothing to run!

Some home-brewed ROM images can be downloaded from http://www.zophar.com. To find commercial games, you could try a web search engine and some imaginative use of search strings. Please note, it is illegal in most countries to have commercial ROM images without also owning the actual SNES ROM cartridge.

Getting Started

Launch Snes9x from the Windows Start menu if you used the package install method; alternatively use Windows explorer to locate the directory where you un-zipped the snes9x.exe and the fmod.dll files and double-click on the snes9x.exe executable. You could create a shortcut to Snes9x and drag that icon out onto your desktop.

Loading Games

Use the Open option from the File menu to open the ROM load dialog. The dialog allows you to browse your computer to locate the directory where you have stored your SNES games. Single-click and then press Load to load and start the game.

SNES ROM images come in lots of different formats, depending on the copier device that was originally used to create the image amongst other things. Sometimes Snes9x has trouble auto-detecting the exact ROM format. Try playing around with the values of the combo boxes in the ROM load dialog if the game doesn't appear to work after its been loaded. In particular, games that use the Super FX RISC processor seem to be mostly in an odd interleaved format that Snes9x has trouble auto-detecting; try selecting 'Interleave mode 2' from the load ROM dialog if you have a Super FX game that isn't working.

Game colour System

Snes9x displays the rom information when a rom is first loaded. Depending on the colours used you can tell whether or not a rom is a good working rom, or if it's been
altered or is corrupted.

-White means the rom should be a perfect working copy
-Green means the rom is mode 1 interleaved
-Cyan means the rom is mode 2 interleaved
-yellow means the rom has probably been altered. Either it's a translation, PD rom, hacked,
    or possibly a bad rom. It may also be an overdumped ROM.
-red means the rom the rom is definately hacked and that a new version should be found.
 some ROM Tools such as NSRT can also fix these ROMs.

When asking for help on the Snes9x forums, please list the colour and CRC32 that

is displayed.
This will help to find out what the problem is.

SNES Joypad Emulation

On a real SNES, players controlled games using an 8-button digital joy-pad; on Snes9x you can use your computer's keyboard (the default) or any joystick or controller device supported by Window's DirectInput. The default key mapping for joy-pad 1 is:

'up arrow'	Up direction
'down arrow'	Down direction
'left arrow'	Left direction
'right arrow'	Right direction
'a'	TL button
'z'	TR button
's'	X button
'x'	Y button
'd'	A button
'c'	B button
'return'	Start button
'space'	Select button

The real SNES allowed up to five joy-pads to be plugged in at once via a special adapter, although there are only a few games that actually supported that many players. Having five people crowd around the keyboard would not be much fun, and anyway, all keyboards have a limit on the number of keys that can be pressed simultaneously and still be detected correctly; much better to use multiple joysticks or NetPlay.

Keyboard/Joystick Config

Add support for your joystick and calibrate it using Windows' joystick applet from the Windows control panel before starting Snes9x, then use Joy-pad Configuration dialog in Snes9x to customise the keyboard/joystick to SNES joy-pad mappings. The dialog is easy to use: select which SNES joy-pad you are configuring using the combo box (#1 to #5), click on the text box next to 'UP' and then press the key on the keyboard or button on your joystick that you would like to perform the UP action. The focus will automatically move on to the 'RIGH T'
text box, press the key or joystick button that you want to perform the RIGHT action, and so on until you've customised all the SNES joy-pad buttons.

If you're using the keyboard, you might want to also program four keys for diagonal movement - I'm told it helps with beat-em-up type games. Click on the each of the blue-coloured diagonal text boxes in turn and press a key on the keyboard you would like to perform the action. Using the numeric keypad for direction keys might be useful in this case.

If you are using a joypad you do NOT UNDER ANY CIRCUMSTANCES SET ANYTHING FOR DIAGONAL. That is strictly a keyboard issue.

Alternate Controllers

Many users have had trouble getting the alternate controllers such as the Superscope working correctly. Due to the way these games use the controller port, some games have issues with them. As Snes9x was designed, it was at one time necessary to add a hack to disable problem devices. As of Snes9x 1.40, the

devices are unavailable for selection by default. This created a somewhat odd method of activating the special controllers. Here is how you manage:

First, select the optional controller you want enabled from the Input menu.
The controller is now selectable by pressing '7'.
Load your game and select the controller using 7. If the game does not pick up the controller, reset the game. At some point, this will probably be rewritten.
We do not know when this will occur.

Additional Keyboard Controls

While the emulator is running:

'Escape'	Show/hide the menu-bar.
'Pause'	Pause the emulator
Alt+'Return'	Toggle between full-screen and windowed mode.
'`'	Superscope turbo button.
'~'	Superscope pause button.
'Tab'	Turbo mode (fast forward) - for skipping long intros.
'0'	Toggle H-DMA emulation on/off.
'1'	Toggle background 1 on/off.
'2'	Toggle background 2 on/off.
'3'	Toggle background 3 on/off.
'4'	Toggle background 4 on/off.
'5'	Toggle sprites on/off
'6'	Toggle swapping of joy-pad one and two around
'7'	Rotate between Multi-player 5, mouse on port 1, mouse on port 2 and SuperScope emulation. (need to enable special controllers in the menu first)
'8'	Toggle background layer priorities for backgrounds involved in sub-screen addition/subtraction.
'9'	Toggle transparency effects on and off - only if 16-bit or higher screen mode selected.
Shift+'9'	Toogle mode 7 bi-linear filtering on and off.
'Backspace'	Toggle emulation of graphics window effects on/off.
Shift+'F1-F9 '	Save a freeze game file.
'F1-F9'	Load a freeze game file, restoring a game to an exact position.
Shift+'+'	Increase emulated frame time by 1ms - slowing down the game. (auto-frame skip only)
Shift+'-'	Decrease emulated frame time by 1ms - speeding up the game. (auto-frame skip only)
Shift+'insert'	Toggles turbo on the 'L' button. Note: toggles for all controllers
Shift+'delete'	Toggles turbo on the 'R' button...
Shift+'['	Toggles turbo on the 'select' button...
Shift+']'	
Shift+'home'	Toggles turbo on the 'Y' button...
Shift+'pageup'	Toggles turbo on the 'X' button...

Shift+'end'	Toggles turbo on the 'B' button...
Shift+'pagedown'	Toggles turbo on the 'A' button...

'+'	Increase frame rendering skip rate, making the screen updates more jerky but speeding up the game.
'-'	Decrease frame rendering skip rate, making the game update more smoothly, but potentially slowing down the game. Repeatedly pressing the key will eventually switch to auto-frame skip rate where the rate is dynamically adjusted to keep a constant game play speed.
'F12'	Takes a screenshot

Several of the menu items also have keyboard accelerators; browse the menus to see what they are.

Joystick Support

Configure and calibrate your joystick/joy-pad using Windows joystick applet in the control panel BEFORE starting Snes9X, then use Snes9X's joy-pad config dialog available from the Options menu to map your joystick/joy-pad's buttons to the emulated SNES joy-pad(s) buttons. See 'Keyboard/Joystick Config' above for details.

Game Saving

Many SNES games could take a very long time to complete from start to finish so they allowed your progressed to be saved into RAM fitted inside the game pack; the RAM contents were backed up by a battery when the SNES was switched off or the game removed. Snes9x simulates this by saving the contents of the emulated battery-backed RAM into a file when you load a new game or exit Snes9x. The file is then automatically re-loaded the next time you play the game.

Snes9x also provides freeze-files; these are files that saves a game's position at any point in the game, not just at predefined places chosen by the game's designers - ideal for saving your game just before a tricky bit! Snes9x provides 9 save slots; during a game, press Shift + F1 to F9 to save a game, and just F1 to F9 to load it again later.

Freeze game files and Save-RAM (S-RAM) save files are normally written to and read from the folder where the ROM image was located, but sometimes this is not desirable or possible, especially if its a CD-ROM, which is of course read-only! You can change the folder where Snes9X saves and loads S-RAM and freeze-files using the Settings Dialog, available from the Options menu.

Snes9x uses its own unique format for freeze-files, as does ZSNES, but Snes9x can also load ZSNES format freeze-files. Just copy the ZSNES freeze files into your save directory and, if the native format Snes9x freeze file doesn't exist (<rom image name>.00X where X is a digit), Snes9x will try to load the corresponding ZSNES freeze file instead (<rom image name>.zsX where X is a 't' or a digit). When you freeze a game position after loading a ZSNES format freeze file, Snes9x will save it in native Snes9x format.

Netplay Support

This support should currently be considered beta. Netplay support allows up to five players to sit in front of different computers and simultaneously play the same game, or just watch someone else play a game. All the computers have to be connected to a network that allows TCP/IP traffic to flow between them; this includes a local Ethernet-style network, a modem connection to another machine, a Windows direct-cable connection, or, if you're lucky and have short ping times, the Internet.

Its currently easier if you use Snes9x in windowed mode while using Netplay, mainly because Netplay currently displays status information in the window's title bar, and it might be necessary to setup a separate chat application so you can talk to the other players when deciding what game to play next.

One machine has to act as a server which other players (client sessions) connect to. The 'master' player, player 1, uses the server machine; the master decides what game to play. The server machine should be selected to be the fastest machine on the fastest connection in the group taking part due to the extra functions it has to perform.

Load up a game, then select the 'Act as server' option from the Netplay menu to become a Netplay server; the 'network', in whatever form it takes, will need to be initialised, if necessary, before you do this. Then just wait for other players to connect...

Clients connect to the server using the 'Connect to server...' dialog, again available from the Netplay menu. Type in the IP address or host name of the machine running the Snes9x server session and press OK. The first remote client to connect will become player 2, and so on. Start Menu->Run->winipcfg will tell you your current IP address, but note that most dial-up ISPs will allocate you a new IP address each time you dial in.

If the server has the 'Send ROM Image to Client' option checked, it will send the client a copy of the game it is currently playing; don't enable this option when using a slow network - sending 4Mbytes+ to several clients will takes ages when using a modem! If the option is not checked the server will request the client loads up the correct game first before joining the game.

Once the client has got a copy of the game the server is playing, the server will then either send it S-RAM data and reset all players' games if the 'Sync Using Reset Game' option is checked, or send it a freeze file to get the new client in sync with the other player's progress in a game.

If the master player loads a different game, the server will either automatically send remote clients a copy, or request that they load the game. If the master player loads a freeze file, the server will automatically send that to remote clients as well.

Client sessions must be able to keep up with the server at all times - if they can't, either because the machine is just too slow, or its busy, the games will get out of sync and it will be impossible to successfully play a multi-player game...

...To make sure this doesn't happen, don't move the Snes9x window unnecessarily and don't use Ctlt+Alt+Del to display the task manager while playing. Also stop any unnecessary applications and as many background tasks as possible. Even something as simple as a text editor might periodically write unsaved data to the disk, stealing CPU time away from Snes9x causing it to skip a frame or delay a sound effect; not a problem for most games, but the Bomberman series (the best multi-player games on the SNES) sync the game to sound samples finishing. Turning off 'Volume envelope height reading' from the Sound Options

dialog might help with this problem.

Cheat Support

Use the Cheat Code Entry and Editor dialog from the Cheats menu to enter Game Genie or Pro-Action Reply cheat codes. Cheat codes allow you to, surprisingly, cheat at games; they might give you more lives, infinite health, enable special powers normally only activated when a special item is found, etc.

Many existing Game Genie and Pro-Action Reply codes can be found at: http://vgstrategies.about.com/library/ggn/bl_ggnsnes.htm?once=true&

Type in a Game Genie or Pro-Action Reply code into the "Enter Cheat Code" text edit box and press <Return>. Be sure to include the '-' when typing in a Game Genie code. You can then type in an optional short description as a reminder to yourself what function the cheat performs. Press <Return> again or click the Add button to add the cheat to the list.

Note that the Add button remains insensitive while "Enter Cheat Code" text edit box is empty or contains an invalid code. The cheat code is always translated into an address and value pair and displayed in the cheat list as such.

Beware of cheat codes designed for a ROM from a different region compared to the one you are playing or for a different version of the ROM; the source of the cheats should tell you which region and for which version of the game they were designed for. If you use a code designed for a different region or version of your game, the game might crash or do other weird things.

It is also possible to enter cheats as an address and value pair; some users have requested this functionality. Type in the address into the "Address" text edit box then type the value into the "Value" text edit box. The value is normally entered in decimal, but if you prefix the value with a '$' or append a 'h' then you can enter the value in hex.

Double-clicking on an cheat line from the list in the dialog or clicking on the "En" column, toggles a individual cheat on and off. All cheats can be switched on and off by checking and unchecking the "Apply cheats" item from the Cheat menu.

Selecting a cheat from the list causes its details to be filled into the text edit boxes in the dialog box; the details can then be edited and the Change button pressed to commit the edits. Note that the "Enter Cheat Code" text edit box always redisplays the cheat code as a Pro-Action Replay code regardless of whether you originally entered it as a Game Genie or Pro-Action Replay code.

Selecting a cheat from the list then pressing the Delete button permanently removes that cheat.

Cheats are saved in .cht files stored in the Freeze File Directory and are automatically loaded the next time a game with the same filename is loaded. The format for the .cht files is the same format as used by the other excellent SNES emulator, ZSNES.

Snes9X also allows new cheats to be found using the Search for New Cheats dialog, again available from the Cheats menu. The easiest way to describe the dialog is to walk through an example.

Cheat Search Example

Lets give ourselves infinite health and lives on Ocean's Addams Family platform game:

Load up the game, keep pressing the start button (Return key by default) to skip past the title screens until you actually start playing the game. You'll notice the game starts with 2 health hearts and 5 lives. Remember that information, it will come in useful later.

Launch the cheat search dialog for the first time, Alt+A is its accelerator. Press the Reset button just in case you've used the dialog before, leave the Search Type and Data Size radio boxes at their default values and press OK.

Play the game for a while until you loose a life by just keep walking into baddies, when the game restarts and the life counter displays 4, launch the cheat search dialog again but this time press the Search button rather than Reset. The number of items in the list will reduce, each line shows a memory location, its current value and its previous value; what we're looking for is the memory location where the game stores its life counter.

Look at address line 7E00AC, its current value is 4 and its previous value was 5. Didn't we start with 5 lives? Looks interesting...

Note that some games store the current life counter as displayed on the screen, while others store current number of lives minus 1. Looks like Addams Family stores the actual life count as displayed on the screen.

Just to make sure you've found the correct location, press OK on the dialog, and play the game until you loose another life. Launch the search dialog again after the life counter on screen has been updated and press the Search button. Now there's even fewer items in the list, but 7E00AC is there again, this time the current value is 3 and the previous value was 4. Looks very much like we've found the correct location.

Now that we're happy we've found the correct location, click on the 7E00AC address line in the list and then press the Add Cheat button. Another dialog, Cheat Details, will be displayed. Type in a new value of say 5, this will be number of lives that will be displayed by the lives counter. Don't be greedy; some games display a junk life counter or might even crash if you enter a value that's too high; Snes9X keeps the value constant anyway, so even if you do loose a life and life counter goes down by one, less than 20ms later, Snes9X resets the counter back to the value you chose!

If the memory location you add a cheat on proves to be wrong, just go to the Cheat Code Editor dialog and delete the incorrect entry.

Now lets try and find the Addams Family health counter. While two hearts are displayed on the screen, visit the cheat search dialog and press the Reset button followed by OK. Play the game until you loose a heart by touching a baddie, then visit the cheat search dialog again.

Press the Search button to update the list with all memory locations that have gone down in value since the last dialog visit. We're going to have to try and find the heart memory location now because there were only two hearts to start with.

Look at address line 7E00C3, its current value is 1 and its previous value was 2. Scrolling through the list doesn't reveal any other likely memory locations, so lets try our luck. Click on the 7E00C3 line, press the Add Cheat button and

type in a new value of say 4 into the dialog that appears and press OK. Press OK on the Search for New Cheats dialog to return to the game.

At first sight it looks like 7E00C3 wasn't the correct memory location because the number of hearts displayed on screen hasn't gone up, but fear not, some games don't continually update health and life displays until they think they need to. Crash into another baddie - instead of dying, the number of hearts displayed jumps up to 4! We've found the correct location after all!

Now every time you play Addams Family you'll have infinite lives and health. Have fun finding cheats for other games.

Super FX

The Super FX is a 10.5/21MHz RISC CPU developed by Argonaut Software used as a game enhancer by several game tiles. Support is still a little buggy but most games work very well, if a little slowly. Released SNES Super FX games included Yoshi's Island (best single-player game on SNES, if you like platform games), Doom, Winter Gold, Dirt Trax FX, StarFox, Stunt Race FX and Vortex. If you're lucky, you might find a copy of the unreleased Starfox 2 image floating around, but its sound code is corrupt and you'll need to disable sound CPU emulation to play it. (NOTE: A new version of Starfox 2 was recently released that is closer to
being a complete rom. It is mostly playable in the emulator but due to it being an
incomplete game it does have many errors. This is not the fault of the emulator)

Lots of Super FX ROM images available are in an odd, interleaved format that I haven't worked out an easy way to auto-detect. If Snes9x detects that a Super FX game crashes (by executing a BRK instruction), it automatically assumes the ROM is in this odd format and de-mangles the ROM and tries to run it again. If your ROM image isn't working, you could try selecting the Interleave mode 2 option on the ROM load dialog before loading the game to help out Snes9x.

S-DD1 and SPC7110

The S-DD1 is a custom data decompression chip that can decompress data in real-time as the SNES DMA's data from the ROM to RAM. Only two games are known to use the chip: Star Ocean and Street Fighter Alpha 2.

SPC7110 is a compression and memory mapping chip. it provides a few extra featur
es,
as well. It functions as an RTC interface, and has a multiply/divide unit that h
as more
precision than the SNES.The SPC7110 is found only in 4 games: Super Power League 4, Far East of Eden Zero, Far East of Eden Zero - Shounen Jump no Shou, and Momotaro Densetsu Happy.

These chips use some unknown compression algorithms, so to actually support the games using these, pre-decompressed graphics packs must be downloaded from http://www.dejap.com, unpacked, and the resultant folder can be selected from th
e
Snes9x options -> GFX Pack Configuration. As of Snes9x 1.42, the S-DD1 can be us
ed
without the graphics packs.

To use the graphics packs for the S-DD1, as of 1.42, you MUST tell Snes9x where the

pack is located. If you do not, it will use on-the-fly decompression. On the oth
er hand,
to use on-the-fly decompression, make sure the game's pack entry is blank.

The SPC7110 still uses packs at all times, and if no pack is defined, it will se
arch the
locations just like ZSNES.

SA-1
====
The SA-1 is a fast, custom 65c816 8/16-bit processor, the same as inside the SNES itself, but clocked at 10MHz compared to a maximum of 3.58MHz for the CPU inside the SNES.

The SA-1 isn't just a CPU, it also contains some extra circuits developed by Nintendo which includes some very fast RAM, a memory mapper, DMA and, several real-time timers.

Snes9X includes emulation of most features of the SA-1, enough to play all SA-1 games I've located so far, these include Mario RPG, Kirby Superstar and Parodius 3.

C4
==

The C4 is custom Capcom chip used only in the Megaman X2 and Megaman X3 games. It can scale and rotate images, draw line-vector objects and do some simple maths to rotate them.

Snes9x's C4 emulation is a direct copy of the ZSNES C4 emulation; Intel-based ports even make use of ZSNES code. Without zsKnight's hard work, Snes9x would not have C4 emulation. Many thanks go to him.

Problems With ROMs
==================

If the emulator just displays a black screen for over 10 seconds after you've loaded a ROM image, then one of the following could be true:

1) You just loaded some random ROM image you've downloaded from the Internet and it isn't even a SNES game or you only downloaded part of the image. Snes9x only emulates games designed for the Super NES, not NES, or Master System, or Game Boy, or <insert your favourite old games system here>.
2) If its a Super FX game, chances are its in interleaved2 format, try switching to "Interleaved mode 2" on the ROM load dialog before loading the game.
3) Someone has edited the Nintendo ROM information area inside the ROM image and Snes9x can't work out what format ROM image is in. Try playing around with the ROM format options on the ROM load dialog.
4) The ROM image is corrupt. If you're loading from CD, I know it might sound silly, but is the CD dirty? Clean, un-hacked ROM images will display [checksum ok] when first loaded, corrupt or hacked ROMs display [bad checksum].
5) The original SNES ROM cartridge had additional hardware inside that is not emulated yet and might never be.
6) You might be using a file that is compressed in a way Snes9x does not underst
and.

The following ROMs are known to currently not to work with any version of Snes9x

:
- All games that use later versions of the DSP-1 (besides DSP-2) chip (e.g. SD Gundam GX,
  Topgear 3000)
- Exhaust Heat2/F1 ROC II (custom co-processor)
- A pair of Shogi games released by Seta, which also use custom coprocessors

Sound Problems

No sound coming from any SNES game using Snes9x? Could be any or all of these:

- If all sound menu options are greyed out, or an error dialog about Snes9x not being able to initialise DirectSound is displayed - then DirectSound could not initialise itself. Make sure DirectX 6 or above is installed and your sound card is supported by DirectX.

  Installing the latest drivers for your sound card might help. Another Windows application might have opened DirectSound in exclusive mode or opened the Windows WAVE device - WinAmp uses the Windows WAVE device by default - in which case you will need to stop that application and then restart Snes9x. It is possible to switch WinAmp to use DirectSound, in which case both Snes9x and WinAmp output can be heard at the same time.

  If your sound card isn't supported by DirectX very well (or not at all) you will have to use FMOD's WAVE output option; but WAVE output introduces a 0.15s delay between Snes9x generating sample data and you hearing it. Select FMOD's WAVE output by choosing the "FMOD Windows Multimedia" sound driver option from the Sound Settings dialog.

- The sound card's volume level might be set too low. Snes9x doesn't alter the card's master volume level so you might need to adjust it using the sound card's mixer/volume controls usually available from the task bar or start menu.
- Make sure your speakers and turned on, plugged in and the volume controls are set to a suitable level.
- You've turned off sound CPU emulation, clicked the Mute button in the sound settings dialog, or set the playback rate to "<No sound>".

General sound problems:
- A continuous, crackling sound or buzz can be heard.

  First make sure it is happening in all games - Snes9x still does have one or two sound emulation bugs that cause the odd pop, crackle and buzz in a few games.

  Once you're happy that it is not just the game you're playing, set the playback rate in the Sound Settings dialog to 22KHz, and uncheck "Stereo" and "16bit playback". Next set both sound buffer length and mix values to 10ms, then try slowly increasing both values until clear sound can be heard. The ideal is that the mix interval and sound buffer length values should be as small as possible. The mix interval value must always be smaller than the sound buffer length otherwise sound data will be lost.

  If your sound card requires larger values, above, say, 40ms, then it might also be necessary to enable the "Generate sample data in sync with sound CPU" option to maintain accurate sound emulation.

  If all else fails, try selecting the "FMOD Window Multimedia" sound driver

option and live with the 0.15s lagged sound that is unavoidable when using the older Windows WAVE sound API.

Once you have clear sound, set the playback rate, 16bit sound and stereo settings to quality you would like - it might be necessary to tweak the sound buffer length and mix interval values again.

- Sound the lagged/delayed compared to on-screen action. If you're using the "FMOD Windows Multimedia" sound driver then delayed sound is unavoidable; otherwise your sound buffer length/mix interval settings are too large - if you can, reduce their values.
- Sound quality is poor on all games. You might have a noisy sound card (usually cheaper sound cards), turning on 16-bit, interpolated sound, sync-sound and/or increasing the playback rate might help.
- Sound seems to have gaps. Using larger sound buffer or mix interval values can cause this; reduce them if you can, or click on the "Generate sample data in sync with sound CPU" option.
- Sound is awful in all games. You might have selected a playback rate/stereo/8-bit/16-bit combination that your sound card can't cope with. Try setting 8-bit (uncheck the 16bit playback option), mono (uncheck the Stereo option) 22KHz from the sound menu and if that cures the problem, try other combinations until you find the best setting that works on your sound card.

If all else fails, try posting a message describing your problem and requesting help on the Snes9X message board at the Snes9X web site, http://www.snes9x.com/

Converting ROM Images

If you have a ROM image in several pieces, simply rename them so their filename extensions are numbered: e.g. game.1, game.2, etc. Then, when loading the ROM image, just specify the name of the first part; the remaining parts will be loaded automatically.

If they are already in the form sf32xxxa, sf32xxxb, etc., you don't even have to rename them; just choose the name of the first part from the ROM load dialog, as above.

Emulation speed

Emulating an SNES is very processor intensive, with its two or sometimes three CPUs, an 8 channel digital sound processor with real-time sound sample decompression and stereo sound, two custom graphics processors, etc.

If you only have a 486 machine, you will need to stick to using only 8-bit graphics and minimal or no sound. Disabling the joystick support will also help.

Full-screen mode is generally faster than windowed mode.

Enabling one of the output image processing modes from the Display Config dialog can slow down overall emulation speed greatly depending on the type of game and video RAM speed. Enabling the stretch image option further reduces emulation speed.

If you want the SNES image to fill your computer screen and want maximum speed, use the 3dfx bi-linear option if you have a Voodoo 3dfx card, or select output image mode as none and check the full-screen and stretch image options.

Lowering the sound playback rate, selecting 8-bit mono sound or turning off interpolated or sync-sound modes, or turning off sound CPU emulation altogether will also help boost emulation speed.

Credits

- Jerremy Koot for all his hard work on previous versions of Snes96, Snes97 and Snes9x.
- Ivar for the original Super FX C emulation, DSP-1 emulation work and information on both chips.
- zsKnight and _Demo_ for the Intel Super FX assembler, DSP-1 and C4 emulation code.
- zsKnight and _Demo_ for all the other ideas and code I've nicked off them; they've nicked lots of my ideas and information too!
- John Weidman and Darkforce for the S-RTC emulation information and code.
- Kreed for his excellent image enhancer routines.
- Nose000 for code changes to support various Japanese SNES games.
- Neill Corlett for the IPS patching support code.
- DiskDude's SNES Kart v1.6 document for the Game Genie(TM) and Pro-Action Replay cheat system information.
- Lord ESNES for some nice chats and generally useful stuff.
- Lee Hyde (lee@jlp1.demon.co.uk) for his quest for sound information and the Windows 95 icon.
- Shawn Hargreaves for the rather good Allegro 3.0 DOS library.
- Robert Grubbs for the SideWinder information - although I didn't use his actual driver in the end.
- Steve Snake for his insights into SNES sound sample decompression.
- Vojtech Pavlik for the Linux joystick driver patches.
- Maciej Babinski for the basics of Linux's DGA X server extensions.
- Alexander Larsson for the GGI Linux port code.
- Harald Fielker for the original sound interpolation code (never used directly due to problems).
- Takehiro TOMINAGA for many speed up suggestions and bug fixes.
- Predicador for the Windows icon.
- Lindsey Dubb for the mode 7 bi-linear filter code and the improved colour addition and subtraction code.
- Anti Resonance for his super-human efforts to help get his fast sound CPU core and sound DSP core working in Snes9x.
- pagefault, TRAC, Dark Force, and others who have donated ideas and/or code to the project.

Gary Henderson
gary@snes9x.com

Updated most recently by: Matthew Kendora