



## INTRODUCTION

SMSPlus emulates Sega's Master System and Game Gear systems. It was originally created by Charles MacDonald and ported to the Nintendo GameCube by The "Genesis Plus" Team. It is an open source emulator and a community project which aims to bring you blast processing into the past. Using this "emulator" you will be able to enjoy all of your classic 8bit games in all of their glory. SMSPlus has been converted from a PC based platform to run on the GameCube's proprietary PPC Gekko processor and features customized code to give you the best gaming experience possible.

This port is based on SMSPlus 1.3 sourcecode but it's also featuring a lot of modifications & additions to improve games compatibility, emulation accuracy and various peripheral supports. See changelog.txt for the whole history.

## FEATURES

### **Accurate & full speed Sega Master System/Game Gear emulation**

- NTSC (60Hz) & PAL (50Hz) timings support
- FM unit emulation (Japanese Master System)
- Savestate support
- SMS BIOS support
- full Overscan area emulation (horizontal & vertical colored borders)
- Blargg's NTSC filter support (NTSC video artefacts emulation)

### **Support for various input peripherals**

- Sega 2-buttons Pad
- Sega Light Phaser
- Sega Paddle
- 3D glasses faking

### **Gamecube/Wii extra features**

- Stereo Sound (@48 kHz)
- 1~2 Players support
- internal Game Database for automatic configuration
- SRAM and SaveState files (on Memory Card & SDCard)
- automatic SRAM/Savestate loading & saving
- support for zipped (.zip) ROM files
- load ROM files from SD, USB (Wii only) or DVD
- original video modes (240p/288p) support
- interlaced (576i/480i) & progressive (480p) video modes support
- Wiimote, Nunchuk & Classic Controller support

## CREDITS

### SMS Plus core

- original emulation code by [Charles Mac Donald](#)
- additional code (core, extra features, compatibility fixes,...) by [Eke-Eke](#)
- Z80 and YM2413 cores by the [M.A.M.E](#) team
- alternate YM2413 core by Mitsutaka Okazaki
- SN76489 core by [Maxim](#)
- NTSC Filter by [Shai Green](#)
- [zlib](#) by Jean-Loup Gailly and Mark Adler
- thanks to [SMS Power 's Wiki](#) and forum members for their technical help

### Gamecube/Wii port

- original Gamecube's port by softdev
- additional features and Wii port by Eke-Eke
- graphical interface and icon design by [brakken](#)
- [libFAT](#) by Chism, ported to libogc by Sven Peter (svpe) & wintermute
- [wiiuse](#) library by Michael Laforest (para), ported to libogc by shagkur
- [libDI](#) by Erant
- [libOGC](#) by winterMute, shagkur and all contributors
- [devkitpro](#) & [devkitPPC](#) by winterMute

*additional thanks to*

- *softdev for all his great work and inspiration*
- *tmbinc for having made Gamecube homebrew a reality*
- *Twiiizer team for all the work they are doing for Wii homebrew*
- *brakken & Tehskeen's forum members for their feedback and support*

## COMPILE THE SOURCECODE

According to the GNU status of this project, the source code **MUST** be **made available to anyone as soon as you have modified it and released a binary**.

To recompile the source code, you will need to be familiar with development environment setup. If you don't, you might look at [wiibrew](#) or [tehskeen's forum](#).

First, you will have to download and install the following tools/libraries:

- the last version of Devkitpro and [DevkitPPC](#). Windows user should directly run the [AutoInstaller](#) version
- the CVS version of [libogc](#).
- the last version of [libfat](#). You can also get and compile the current source from [CVS repository](#)
- the last [libDI](#) source code.

Once you are done, grab the current [smsplus-gx](#) source code from SVN (<http://code.google.com/p/smsplus-gx/source/checkout>) then launch compile\_all.bat from **msys** (installed with devkitpro)

## RUN THE EMULATOR

**smsplus\_cube.dol** is the application running in Gamecube mode. They can be loaded on a Gamecube or a Wii (using GC compatible mode) through various methods (Bootable DVD, SDLOAD,...). If you have no idea on how to load a DOL, please go here on follow the available guides: <http://modyawii.tehskeen.com> (Booting Homebrew Section).

**smsplus\_wii.dol** is the application running in Wii mode, using extra features like wiimotes and native SD slot support. They can be loaded on a Wii using either the TP Loader or the Homebrew Channel. See <http://www.wiibrew.org/> and <http://hbc.hackmii.com/> for more information on how to run .dol and .elf files on your Wii. To use it with the Homebrew Channel, simply rename smsplus\_wii.dol to boot.dol and place it, with meta.xml & icon.png, on your sdcard, in the /apps/genplus directory.

## SETUP THE EMULATOR

Supported ROM files format are common .sms and .gg Sega Master System/Game Gear ROM file format. However, .zip files are also supported as long as they contain a valid ROM image file.

ROM files can either be loaded from a SDCARD or from a DVD. You can store them anywhere and it is strongly recommended to use subdirectories and limit the amount of ROM files per directory in order to speed-up the browsing process. According to the device you are using to load ROM files, please consider the following rules:

- **SDCARD or USB Storage** (only for Wii) devices should be formatted in FAT (FAT16 only for Gamecube). The emulator will always be looking in the directory **"/smsplus/roms"** by default so it is recommended to create these directories (on a side note, configuration files are saved in the /smsplus directory) and put your ROM files here. If the directory does not exist, the program will let you browse from the root of the SDCARD.
  - The Gamecube version requires a SD-adapter inserted in one of the two memory card slots (automatically). SDHC device are **not** supported.
  - The Wii version requires that you insert a SDCARD inside the front SD slot (SD-adapter is not supported) or connect an USB storage device. SDHC device should be supported.
- **DVD** should use ISO9660 file format. The emulator starts looking for files from the root directory. The maximal size of the DVD depend on the type of the DVD drive: the Gamecube Mini-DVD drive allows up to 1.35GB of data and the Wii DVD drive allows up to 4.7GB of data (simple-layer).
  - The Gamecube version requires a modchip to be able to read DVD-/R
  - The Wii version does not requires a modchip but you will have to install the DI layer "softmod" separately. If you have a modchip, you might need to install a patched IOS. More information about the DI softmod can be found here: <http://hackmii.com/2008/08/libdi-and-the-dvdx-installer/>

## USE THE EMULATOR

You'll start off with the main introduction screen and after pressing "A" you will be at the main menu.

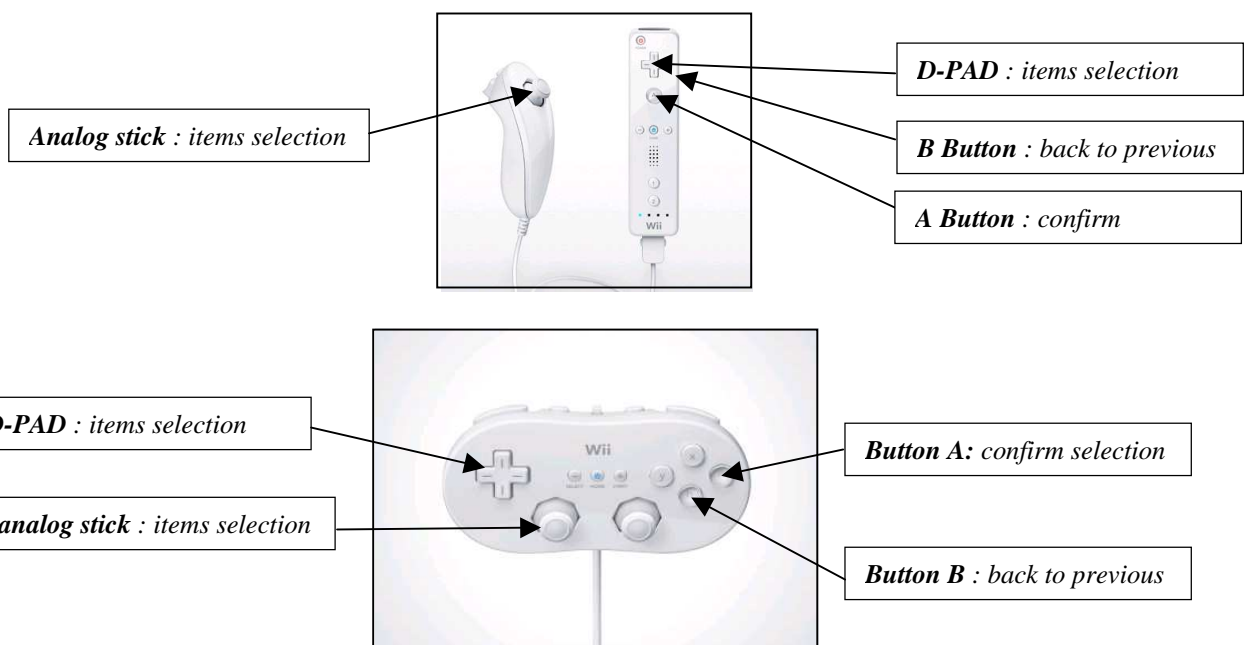


When you are navigating through the menus, the following keys on your Gamecube controller are used:



### Wii version

You can also navigate through the menu using the Wiimote and expansion controller. In the Menu, keys are mapped as the following:



## PLAY GAME

This will takes you into or back to the game. During gameplay, use a Gamecube pad to control.
























### WII version






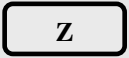







You can also use the Wiimote and expansion controllers. There are **3** possible configurations depending on the type of expansion controller that is inserted when you play a game:

1. WIIMOTE only
2. WIIMOTE + NUNCHUK combination
3. CLASSIC CONTROLLER

Each of three configurations have a default key mapping listed below but can also be reconfigured separately (see “**Configure Inputs**” option) for each players. A maximum of 4 WIIMOTES can be synchronized.

The following table gives you the default mapping for each configuration, dark grey entries aren't reconfigurable.

				
	 		 	 
 <b>PAUSE</b>	START 			
 <b>BUTTON 1</b>				

				
<b>BUTTON 2</b>				
<b>MENU</b>				
<b>SOFT RESET</b>				

**NOTE:** Some games requires specific peripherals, such as PADDLE or LIGHT PHASER to be emulated. The program contains an internal game database and automatically activate emulation of these peripherals.



When the **LIGHT PHASER** is required: you can use D-pad or ANALOG Sticks to move the cursor but the best enjoyable way is to use the WIIMOTE to aim the screen. When using the WIIMOTE, Button A is automatically assigned as the TRIGGER button.



When the **PADDLE** is required, use either D-PAD or ANALOG Sticks to control the game. WIIMOTE pointing can also be used but isn't really precise in this case.

**NOTE:** Soft Reset can also be performed by pressing the Gamecube/Wii RESET button.

## HARD RESET

This should be like switching OFF/ON the POWER button on a real Master System. This will completely reinitialize the genesis virtual machine.

## LOAD NEW GAME

**Load Recent** let you browse a ROM history list with the ten last opened ROM files. This is only available for ROM files previously loaded from SDCARD.

**Load from SD** let you browse the SDCARD.

**Load from USB** let you browse the USB Storage device (Wii only).

**Load from DVD** let you browse the DVD.

**Stop DVD Motor** will stop the DVD motor and the disc from spinning during playtime

Once you have selected an option, a file selection menu should appear. In this new selection menu, the following controls can be used:

### GAMECUBE PAD

- A button : load the selected file
- B button : go up one directory
- Z button : quit the file selection menu
- L/R triggers : go down/up one full page
- Left/Right buttons or Analog stick : display the selected entry's full filename
- Up/Down buttons or Analog stick : select previous/next file

### WIIMOTE, WIIMOTE+NUNCHUK

- A button : load the selected file
- B button : go up one directory
- HOME button : quit the file selection menu
- +/- Buttons: down/up one full page
- Left/Right buttons or Analog stick : display the selected entry's full filename
- Up/Down buttons or Analog stick : select previous/next file

### CLASSIC CONTROLLER

- A button : load the selected file
- B button : go up one directory
- HOME button : quit the file selection menu
- L/R triggers: down/up one full page
- Left/Right buttons or Analog stick : display the selected entry's full filename
- Up/Down buttons or Analog stick : select previous/next file

## SAVESTATE MANAGER

Let you load/save SaveState data from/to the selected device: SaveState files is a feature that does not exist on real hardware and will let you save and restore your progress in ANY games, even those which don't have internal saving feature. The SaveState data can be seen as a snapshot (or "freeze" state) of the current emulation state. Once restored, you will be able to continue your game at the **exact** point where you leaved it.

You can choose the device type. Be sure to set this according to your system configuration before saving/loading files:

#### Notes:

- you can configure the emulator so that it will automatically save SAVESTATE for the current game as soon as you load another game or quit the application. On the same way, SAVESTATE can automatically be restored after you load a game, if the files exist. This feature can be enabled/disabled via the **System Options** menu.
- when using NGC Memory Card in SLOTA, some mounting errors may occur. In this case, remove and insert the Memory Card again before trying to save/load anything or use SLOTB. Be sure to have also enough space on the Card before trying to save something (Freeze State and SRAM files are usually compressed).
- when using FAT device, the directory **/smsplus/saves** is automatically created. The default device is detected during initialization.

# EMULATOR OPTIONS

## DISPLAY Options

**Aspect** let you choose the Display Aspect Ratio:

- ORIGINAL mode automatically set the correct aspect ratio exactly as if you connected a real Master System on your TV. In this mode, the full resolution (720 x 480 or 720x574 pixels) is used to include horizontal and vertical borders around the active display area.
- STRETCH mode let you adjust horizontal and vertical scale values so that the active display fits your TV screen. When using this mode, borders are not emulated.

**Render** let you choose the Display Rendering mode:

- ORIGINAL let you use the original Master System rendering mode: this mode outputs a progressive 240 lines (288 lines for PAL) display.
- INTERLACED vertically scales (using hardware filtering features) the original display to a 480 lines (574 lines for PAL) interlaced display. In this mode, because of the higher resolution, games generally look better than on the real hardware but some artefacts might appear during intensive and fast action.
- PROGRESSIVE switches into Progressive Video Mode (480p), only use this with component cable and a compatible TV.

**TV Mode** let you choose the TV Mode to use:

- 50/60Hz: in this mode, the Gamecube automatically switch between the appropriate 50hz and 60Hz TV modes depending on the Master System current region mode. This makes PAL & NTSC games looking exactly like they did on a real console..
- 60Hz: in this mode, the Gamecube always use a 60Hz (NTSC or PAL60) TV mode, use this if your TV does not support 50Hz.
- 50Hz: in this mode, the Gamecube always use a 50Hz (PAL) TV mode, use this if your TV does not support 60Hz.

**Bilinear Filter** let you enable/disable hardware bilinear filtering. Emulation speed is not affected by this setting, however, when filtering is OFF, some graphic flickering might appear during screen scrolling.

**NTSC Filter** let you enable/disable Blargg's NTSC filters. These are software filters that emulates the native artefacts produced by the NTSC Video Signal (color blending,...). Several predefined modes are available (COMPOSITE, S-VIDEO & RGB) which simulates the associated video signal.

**Borders** let you enable/disable the border colour emulation: when ON, the background colour is used (like on a real console). When OFF, borders are forced to black. When ASPECT mode is set to STRETCH, this option is automatically set to OFF. On the other way, enabling this option automatically set ASPECT mode to ORIGINAL.

**Center X/Center Y** let you adjust the screen position while keeping the display aspect ratio.

**Scale X/Scale Y** let you adjust the display aspect ratio. This option is only accessible when using STRETCH aspect mode.



## SYSTEM Options

**FM Type** let you disable FM emulation (OFF) or choose between two FM emulators (YM2413,EMU2413). Please note that only some Japanese games feature FM music so enabling this will force the system into Japanese mode.

**Country** let you force the region of the game. Let this to AUTO if you want the program to pick automatically the best setting for each game. Some games may be region protected.

**Console** let you force the type of console to be emulated (Master System 1, 2 or Japanese (MARK III), Gamegear...), Let this to AUTO if you want the program to pick automatically the best setting for each game.

**Sprite Limit** let you enable/disable emulation of original hardware limitation that can produce sprite flickering in some games. Disabling this option will reduce Sprite Flickering and improve display rendering.

**Use BIOS** let you enable/disable SMS BIOS support. If you want to use this feature (this is not required to play games), the BIOS rom (not provided) must be renamed as "**BIOS.sms**" and placed in the /smsplus/ directory, on the default FAT device.

**Extra GG** let you enable/disable the hidden display area, normally hidden by the original Game Gear LCD screen edges.

**FREEZE Auto** let you enable/disable automatic *SaveState* loading when a new game has been loaded and autosaving when you quit the emulator or load a new game. This option let you specify the location for the *SaveState* files : FAT (SD or USB for Wii, default device is detected upon startup), MEMCARD (slot A or slot B)

## CONTROLS Options

**Configure Player** let you change the current player input (that should be modified by the options below).

**GAMEPAD** let you modify the way buttons are mapped on the gamecube pad. Just follow the instructions and press the desired buttons when asked.

**WIIMOTE** let you modify the way buttons are mapped on the current wiimote configuration (remember that 3 configurations can be used: WIIMOTE, WIIMOTE+NUNCHUK and CLASSIC CONTROLLER). Just follow the instructions and press the desired buttons when asked.

## RETURN TO LOADER

This allow you to return to the loader used to launch the program, if any (Homebrew Channel, TP Loader, SDLOAD, PSLOAD)

## SYSTEM REBOOT

This will reboot your console (Gamecube Mode) or return to the Wii System Menu