## Introduction



#### What is Gens?

Gens is a very good Sega Genesis/Megadrive, SegaCD/MegaCD, 32X emulator by Stef.

Stef made it open source some years ago.

#### What is KMod?

A mod of the original Gens source code to add development/debugging/hacking features.

It is a replacement to very useful but buggy Genecyst features.

#### What is NOT KMod?

KMod is NOT a new emulator, I used Gens engine.

To be honest it's even worst than original Gens because it slows down it.

#### Who is behind KMod?

My nickname is Kaneda, sometimes KanedaFr.

I'm making some development on Sega Megadrive/Genesis since '98.

Some of my known demos/games are Ooga Booga and Old Skill.

I also made a lot of tech demos and tools.

## Copyright ©2007 Kaneda

# History

#### Version 0.7b

Better watchers with structures handling

Version 0.7b

Better watchers

#### Version 0.7

FIX: create file and directories at ROM place when 'opened from..'

(Pascal/TmEE)

FIX: bad redraw when shadow/highlight and layer hide (GrayLight)

FIX: wrong redraw of S68K PRam adresses in RAM mode (Fonzie)

FIX: screenshot 32X not working

FIX: pause 32X unperfect (since KMod 0.3!)

changed debug font for W98 user

better PSG debug window by Edge

DAC frequency info

MSH2 ROM/RAM/cache viewer and linked disassembly (thx Fonzie)

SSH2 ROM/RAM/cache viewer and linked disassembly (thx Fonzie)

32x registers

32x VDP(s)

CPUs debug windows are on the same template

Spy CD Bios call

Autolog message to file, so no more limited to memory

add tfi dump (Shiru)

5th pal for debug (GrayLight)

#### Version 0.6

FIX: bad redraw when shadow/highlight and layer hide

FIX : correct DAC value

FIX: screenshot takes the message too

Show/Hide 32X layer

Z80 memory viewer and linked disassembly (no more 'current' PC)

M68K ROM/RAM viewer and linked disassembly (no more 'current' PC)

S68K PRAM/WRAM viewer and linked disassembly (no more 'current' PC)

debug.s/.h updaded (warning! func name updated too...sorry for that, but it's now more 'pro')

Dump CD Word RAM to BMP

Change scroll button with true scrollbar on CD GFX Change scroll button with scrollbar on VDP (and remove up/down bitmap resource)

#### Version 0.5

FIX: bad 68k memory dump (swap)

FIX : bad MCD reg list FIX : rename s68k buttons Spy unpaused Z80 access

Register 29: more controls than only Pause

Debug.s updated GMV record/play

GMV Tools to compress GMV for use in demos replay

Pause on menu

Render frame per frame

Auto screenshot

#### Version 0.4

FIX : memory fault on sprite dump (thx RedAngel)

FIX: bad tile selection

FIX: no tiles refresh while paused (thx tomman)

Spy/Message on separate window

debug CD 68k debug CD CDC

debug CD Reg

debug CD GFX

debug VDP Registers more readable

spy DMA to hack gfx

spy DMA limit error (thx CMD)

dump M68K rom (interesting when CD or 32X game) dump S68K rom (interesting when CD or 32X game)

Register 31: Timer Register 30: Message Register 29: Pause Gens

special registers calls in zip file available in Debug...

#### Version 0.3

FIX: on pause mode, if debug window called, no redraw (black screen)

FIX: YM2612 now updated while game paused

FIX: pal red arrow now centered
FIX: correct YM2612 TimerB value
VDP redraw optimized (thx Fonzie)
Sprite redraw optimized
'blue pause' optionnel (Fonzie req)
dump VDP tiles to bmp
dump sprite to bmp
dump ym2612 instr (y12 or ff)
supp key to delete a watcher (Varcies req)
ins key to delete a watcher
active/desactive any pal (color or black)
sprite zoom
tile zoom

#### Version 0.2

Debug config window special config file (gens\_kmod.cfg) autoload watchers (thx Fonzie!) spy log (<gamename>.log) spy strange register data (known as CharlesMD patch) spy full sprite limit log/watcher name ok when loading from history rom (fix) spy bad read (68K Byte, 68K Word) spy bad write (68K Byte, 68K Word)

#### Version 0.1

scroll tiles by line
pal indicator (red arrow)
select current pal in VDP debug by click
show/hide layers (known as Genecyst patch)
save/load watcher (<gamename>.wch)
create watchers directory on init

#### **Version 0.0 - private release**

replace Debug == X by Debug == DMODE\_XXXXX for easier reading call debug = screen minimize (if fullscreen) change CPU->Debug and Options menu debug M68K debug z80

debug VDP (pal/tile) debug VDP Registers debug sprites debug ym2612 debug PSG watchers

# Copyright ©2007 Kaneda

# Changes

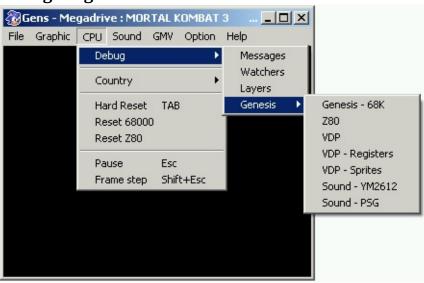
When I started Gens KMod, my main goal was to have a separate window for each of the debug screen while the game is playing.

Since the code was very well documented, I was able to add several non existing features.

Check them all.

Please note that the way Gens was written doesn't allow me to add breakpoints and trace features without huge rewrite, so don't expect these features.

#### **Debug Megadrive/Genesis Game**



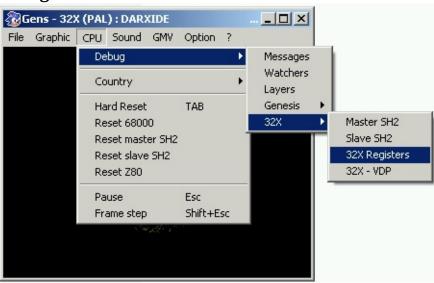
- Genesis 68K status
- <u>Z80 status</u>
- <u>VDP</u>
- VDP Registers
- VDP Sprites
- YM2612 status
- PSG status

#### Debug MegaCD/SegaCD Game



- CD 68K status
- CD Registers
- CDControler
- CD Word RAM

## **Debug 32X Game**



- Master SH2 status
- Slave SH2 status
- 32X registers
- 32X VDP

## **Development features**

- New register 31: Timer

- New register 30 : Message
- New register 29 : Take Gens control
- Watchers
- <u>Spy</u>

## Misc changes

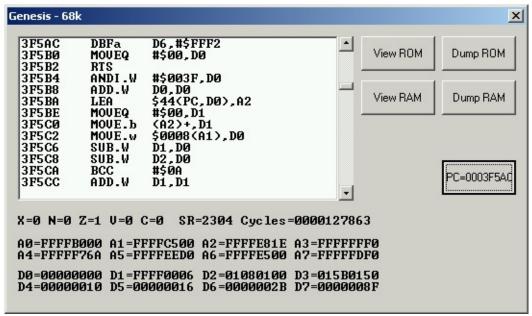
- Configuration
- Message window
- <u>Layers</u>
- Extended Features
- <u>GMV</u>

# Copyright ©2007 Kaneda

# Genesis

# Copyright ©2007 Kaneda

# Genesis 68K



The current Genesis 68K status is now on a separate window.

You can now view any disasm code at any address and not current code, which was unuseful.

Use it only for its memory features.

## **Dump ROM**

Dump the full ROM in binary format.

This feature is interesting while running a MegaCD/SegaCD or 32X games, else it works like a SMD to BIN convertor

#### View ROM / View Disasm

Switch between hexaview and disasm view (see picture below) of the ROM.

#### View RAM

View RAM dump.

### **Dump RAM**

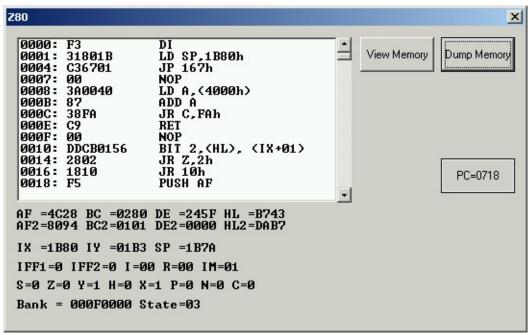
Dump the full RAM in binary format.

Usefull to save multiple RAM states and compare them or check memory status

#### PC = xxxxx

In Disasm view, jump to current PC

# Copyright ©2007 Kaneda



The current Z80 status is now on a separate window.

Use it mainly for its memory features.

#### View Memory/Disasm

Switch between hexaview and disasm view (see picture below) of the Z80 memory.

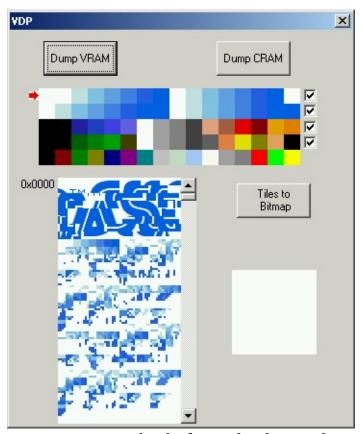
## **Dump Memory**

Dump the Z80 memory in binary format. Usefull to check Z80 sound driver

#### PC = xxxxx

In Disasm view, jump to current PC

## Copyright ©2007 Kaneda



One of the most important part to check if your loading is done or get help to hack

The picture at the top is the 4 palettes of the Genesis + 1 for us, to spy tiles even when no palette is loaded.

You can choose by click the current one for the VDP debug

With the checkboxes, you can also active/desactive the pal you don't want, so you can see on screen what pal any tile/sprite use.

The picture at the left is the current VRAM, drawn with the current select pal.

You can scroll it to see all the tiles you want

You can select a specific tile by click to see it in details.

Warning, for refresh speed need, all the tiles aren't refreshed every frame.

The picture at the right is the zoomed selected tile (0 by default)

### **Dump VRAM**

Dump the full Video RAM in binary format

# **Dump CRAM**

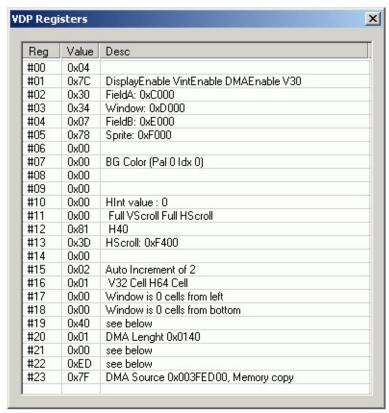
Dump the full Colors RAM in binary format

## Tiles to Bitmap

Save the full Video RAM in a Bitmap file Yes, you can easily rip 8x8 tiles

## Copyright ©2007 Kaneda

# VDP Registers



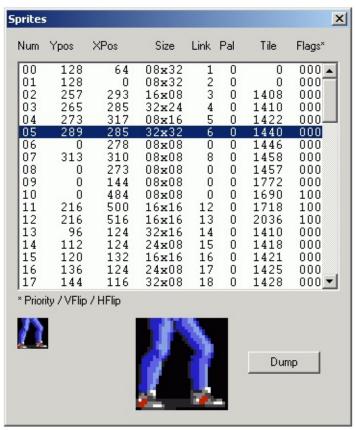
Check the status of the VDP registers.

Nothing more, you should know what the Genesis' registers are.

Please note I didn't add Charles more accurate registers information, but I hope to do it ...some day

Copyright ©2007 Kaneda

# **VDP** Sprites



Check the sprites list.

*Num*: sprite number

*YPos/XPos*: position in pixels *Size*: size in pixels (32x32 max)

*Link*: next sprite (see Sega2.doc for more details)

Pal: pal used

*Tile*: first tile used

Flags: Prioity / Vertical Flip / Horizontal Flip

Select by click the sprite you want to see

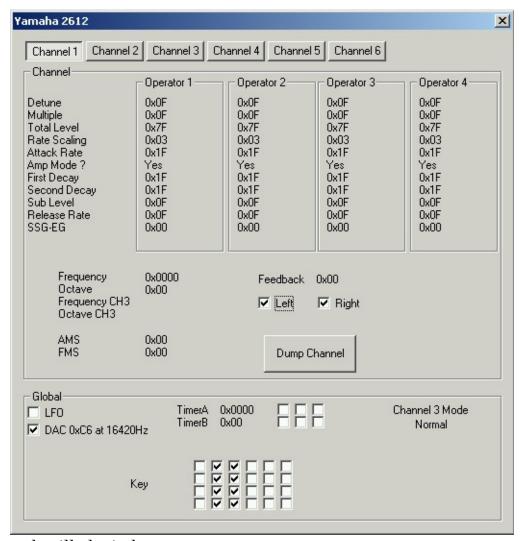
The picture at the left is the sprite in original size. The picture at the right is the zoomed sprite.

## **Dump**

# Dump the selected sprite in a Bitmap file Yes, you can easily rip sprites

# Copyright ©2007 Kaneda

# YM2612

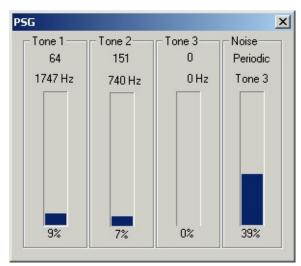


The most detailled window.

Let you check each channel of the YM2612 sound chip See Sega2.doc for more details.

## **Dump Channel**

Dump the channel info in FF (Uchuusen's format), TFI (TFM'ucker instrument) or Y12 (Kaneda's format)



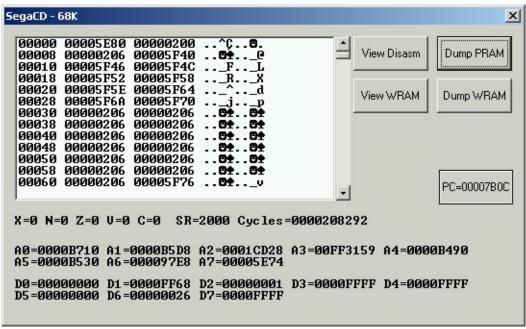
Check the value and volume of each PSG channel

# Thanks to Edge who rewrite this window

Copyright ©2007 Kaneda

# Sega CD

# Copyright ©2007 Kaneda



The current CD 68K status is now on a separate window.

You can now view any disasm code at any address and not current code, which was unuseful.

Use it only for its memory features.

## **Dump Prgm RAM**

Dump the full Program RAM in binary format.

## View Prgm RAM / View Disasm

Switch between hexaview and disasm view of the Program RAM.

#### **View Word RAM**

View Word RAM (mode 2) dump.

## **Dump Word RAM**

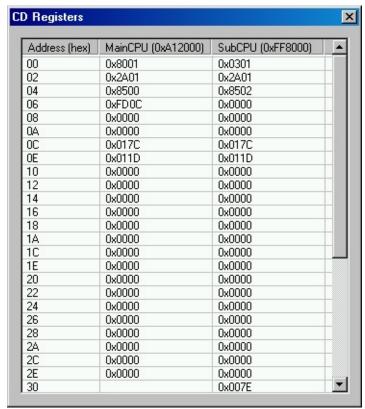
Dump the full Word RAM (mode 2) in binary format.

Usefull to save multiple RAM states and compare them or check memory status

#### PC = xxxxx

In Disasm view, jump to current PC

# CD Registers



Check the status of the CD registers.

Nothing more, you should know what the CD registers are.

I'm currently note a CD pro myself so perhaps there are errors on this screen.

Copyright ©2007 Kaneda

# CD Controler



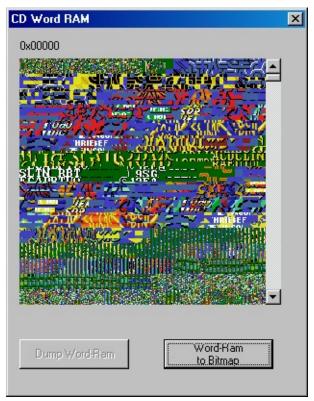
Check the status of the CDC.

Nothing more, you should know what the CD controler is.

I'm currently note a CD pro myself so perhaps there are errors on this screen.

Copyright ©2007 Kaneda

# CD Word RAM



See the tiles in the Word RAM.

Use the palette selector of the **Genesis VDP** to change color.

## **Word-RAM** to Bitmap

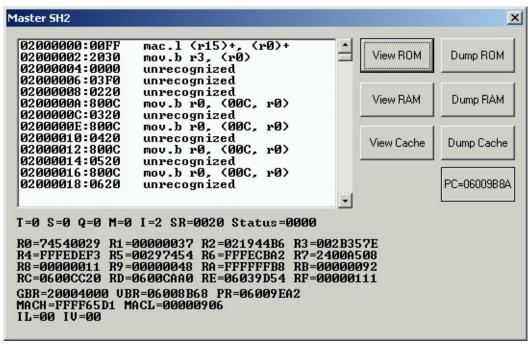
Save the Word RAM as bitmap, using the current selected palette.

# **Dump Word-RAM**

See CD 68k

## Copyright ©2007 Kaneda

# Copyright ©2007 Kaneda



The current Master SH2 status.

#### View ROM / View Disasm

Switch between hexaview and disasm view (see picture below) of the ROM.

### **Dump ROM**

Dump the full ROM in binary format.

#### **View RAM / View Disasm**

Switch between hexaview and disasm view (see picture below) of the RAM.

## **Dump RAM**

Dump the full RAM in binary format.

#### **View Cache**

View the cache

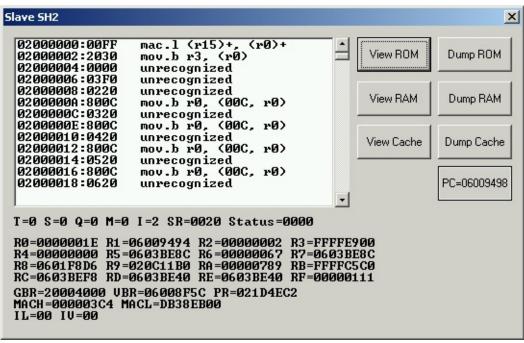
#### **Dump Cache**

Dump the full cache in binary format.

PC = xxxxx

# Jump to current PC (switch to RAM or ROM if needed)

# Copyright ©2007 Kaneda



The current Slave SH2 status. (same window as Master SH2)

#### View ROM / View Disasm

Switch between hexaview and disasm view (see picture below) of the ROM.

## **Dump ROM**

Dump the full ROM in binary format.

#### **View RAM / View Disasm**

Switch between hexaview and disasm view (see picture below) of the RAM.

## **Dump RAM**

Dump the full RAM in binary format.

#### **View Cache**

View the cache

#### **Dump Cache**

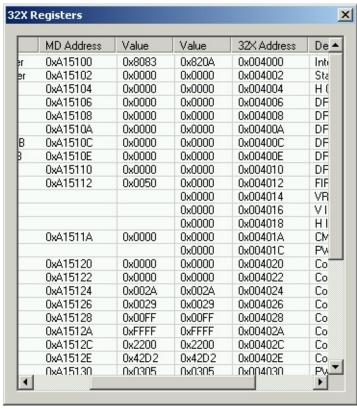
Dump the full cache in binary format.

PC = xxxxx

# Jump to current PC (switch to RAM or ROM if needed)

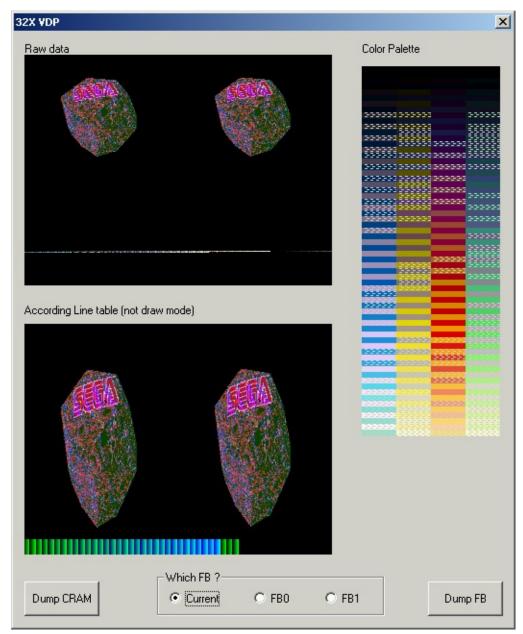
# Copyright ©2007 Kaneda

# 32X registers



Check the status of the VDP registers. You're able to see Genesis and 32X side registers

Copyright ©2007 Kaneda



The complex 32X VDP window

#### Raw data

a raw write of the selected Framebuffer (without line table)

## **According Line table**

the selected Framebuffer drawn like the line table define it

#### **Color Palette**

for 256 colors games, the colors used. the strokes help you to see 'priority'

#### Which FB?

Select the FB you want to spy

### **Dump CRAM**

dump the raw data of color ram (256 colors mode)

### **Dump FB**

dump raw data of the full frame buffers

## Copyright ©2007 Kaneda

# Dev : Register 31

# Register 31 : Timer

### ! WARNING: this register doesn't work during VInt!

As any added register, the original Megadrive/Genesis should skip them (as written in the Dev Kit documentation)

#### Timer register

start/reset and output a timer (based on m68k cycles)

Value is *sr000000* where

s=0 Counter output on Message window

s=1 Counter start r=0 <undefined> r=1 reset counter

I made it for optimizing issue : you can know how cycles a call take and optimize it.

## Copyright ©2007 Kaneda

# Dev : Register 30

# Register Message 30

As any added register, the original Megadrive/Genesis should skip them (as written in the Dev Kit documentation)

#### Message register

output any text on the Message window

Value is a printable char or 0 for output Use a loop to input a full string but be aware Message register will automatically ouput a 255 chars string and start a new one

#### Copyright ©2007 Kaneda

# **Register Gens control 29**

As any added register, the original Megadrive/Genesis should skip them (as written in the Dev Kit documentation)

### **Gens control register**

take control of Gens throught your rom

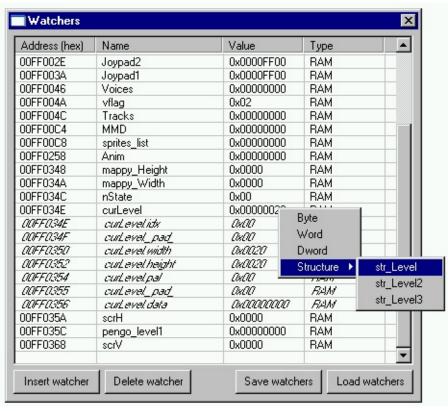
Value is one of these control codes

value is one of these control codes		
0x00	pause	
0x01	show debug 68k window	
0x02	show debug z80 window	
0x03	show debug VDP window	
0x04	show debug sub68k window	
0x05	show debug CDC window	
0x06	show debug CD Gfx window	
0x07	show debug 32X main window	
80x0	show debug 32X sub window	
0x09	show debug 32X VDP window	
0x0A	show debug VDP register window	
0x0B	show debug Sprite window	
0x0C	show debug YM2612 window	
0x0D	show debug PSG window	
0x0E	show watchers window	
0x0F	show Layers window	
0x10	show message window	
0x11	show debug CD register window	
0x50	dump 68k ram	
0x51	dump Z80 ram	
0x52	dump S68k ram	
0x60	take screenshot	
0x62	start record gmv	
0x63	stop record gmv	
0x64	start gmv	
0x65	stop gmv	

# $0x00/\mbox{\sc Pause}$ is the more useful one : pause your game at an important point and check what you want

# Copyright ©2007 Kaneda

# Dev : Watchers



Spy any value of the memory space.

It's a great feature for developpers because, with the help of <a href="mailto:nm2wch">nm2wch</a> tool, you can generate watchers list from source file!

For hackers, it lets you spy memory space for points, life, score...

I also added structures support.

For that you need to add a .str file with 1 or more structure definition.

In the screenshot, str\_Level is defined like this:

```
STRUCT
           str Level
   CHAR
             idx
             _pad_
   CHAR
   SHORT
             width
   SHORT
             height
   CHAR
             pal
   CHAR
             _pad_
   LONG
             data
END
```

so, any structure must start by 'STRUCT' keyword followed my struct name (20char max)

then you add any element you want using 'CHAR', 'SHORT' or 'LONG' keywords followed by element name (20 char max)

it's a good use to add the 'END' keyword but it's optionnal.

Any other keywords or strings (like // or others comments keywords) are ignored.

On "Watchers" list view, row in italic are sub element of a watcher, you can't change its type and if you delete it, you delete the watcher (and all its sub elements).

A side note on the 'CHAR' type, like you can see on the sample, I added to \_pad\_ element because M68K can't read word of odd address.

It's interesting: this 10 bytes structure uses in fact 12 bytes in memory!

#### Address (hex)

Input the address to watch in hexadecimal

On right click, a popupmenu allows you to jump to memory view at this place (RAM only).

Change its value on double click.

#### Name

Name of the watcher (if from a nm2wch convertion).

You can't change it yourself (sorry, need to find how first, WinAPI doesn't handle thisby default)

#### Value

Value of the watcher, in select format.

On right click, a popupmenu let you choose the format of output.

For complex imported watchers, you can choose one of your defined structure.

### Туре

Tell you what you're looking for (up to 0x00400000 ROM

up to 0x00800000	Sega 1
up to 0x00A00000	Sega 2
up to 0x00A10000	Z80
up to 0x00A11000	IO
up to 0x00A12000	Control
up to 0x00B00000	Sega 3
up to 0x00C00000	Sega 4
up to 0x00E00000	VDP
from 0x00FF0000	RAM

It's dangerous, but possible, to watch between 0x00E00000 and 0x00FF000, so you'll see !! WARNING !!

#### **Insert watcher**

At a new watcher at 0x00000000 Edit the address you want to watch

#### **Delete watcher**

Delete current highlighted watcher

#### **Save watchers**

Save all the watchers (see <u>Autoload watcher option</u>)

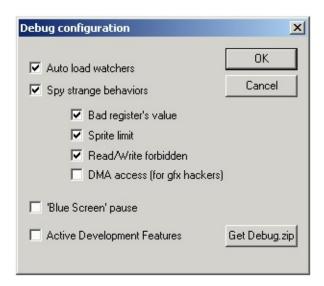
Structure information ARE NOT saved to avoid strange behavior if you delete, add or change order of structures definition in the .str file.

#### **Load watchers**

Load a new set of watchers (the current ones are deleted)

### Copyright ©2007 Kaneda

# Dev: Spy



Enable or not throught the configuration, the spies let you know if you made code error or if a game is full of bugs
I added a special one which will help you to hack game too

#### Bad register's value:

Spy if a bad value is given to a VDP register.

Check errors or unknown register value

### Sprite limit:

Spy if the 80 sprites per screen limit is reached.

Check errors or sprites available only on emulator.

## Read/Write forbidden:

Spy bad memory access.

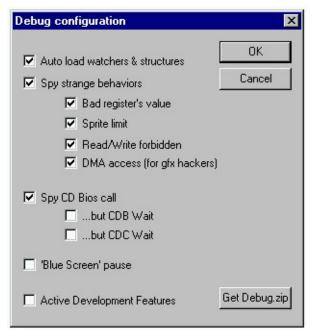
Check errors and bugs

#### DMA access:

Spy ANY DMA.

Usefull to find where are stored tiles, colors or scroll value

# Misc : Configuration



You can open this KMod's configuration throught *Options>Debug...* 

#### Auto load watchers & strucutres

Load watchers with same name as game when loading game.

GensKMod first looks for a wacher (.wch) in current game folder. If not found, it then looks in GensKMod's 'watchers' folder.

Same thing for structures (.str)

### Spy strange behaviors

Enable code spies, the result is wrote on the Message box

### Spy CD Bios call

Dump all CD Bios call on the <u>Message</u> box (you can desactivate the often called functions)

### Blue Screen pause

Pause in blue?

Just disable this Gens feature : I didn't want to remove it so I made it optionnal. It's now more easy to capture a paused screen

### Active Development Features

Active or not Gens only code like specials registers and Gens control by the rom

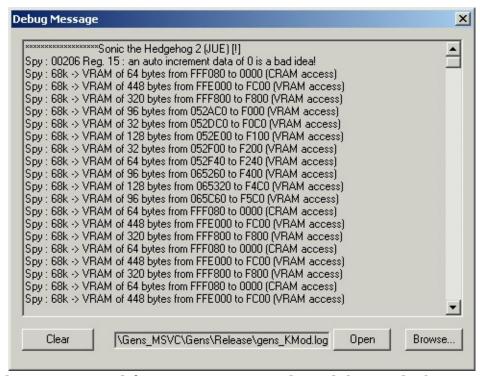
itself

# **Get Debug.zip**

Download the last version of the Gens-only debugging code for your demos and games

# Copyright ©2007 Kaneda

# Misc : Message



If Spy and/or Dev special features are activated, read the results here or in the log file (delete it time to time)

#### Clear

Only a maximum number of lines is wrote so clear times to times

### **Open**

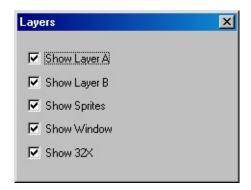
Open the log file with notepad

#### **Browse**

Select the path and filename of the log file

## Copyright ©2007 Kaneda

# Misc : Layers



Select which field(s) you want to be drawn. Interesting for screenshot without sprites for example.

I missed this Genecyst's feature so much!!!

### Copyright ©2007 Kaneda

# Misc : Extended Features

Pause on menu : CPU>Pause

A very little mod, for people who don't know how to pause Gens or/and don't read the manual

Pause update : CPU>Frame Step

Using Shift-Esc or menu, you can update frame per frame, to see all the details drawn each frame.

### Screenshot update: Graphic> Screenshot each frame

If you want to capture each frame.

Easier than CPU>Frame step then Graphic>Screenshot then CPU>Frame step .... Used to record animation.

# Copyright ©2007 Kaneda

# Misc : GMV

# **Gens MoVie**

A GMV is not a video file, it's a record of all the inputs while playing. You can record it then replay it

See NesVideos for more details

#### **GMV>GMV Play**

Load the current state then start to play the GMV file (same name than rom loaded)

#### **GMV>GMV Record**

Save the current state then start to record the GMV file (same name than rom loaded)

#### **GMV>GMV Stop**

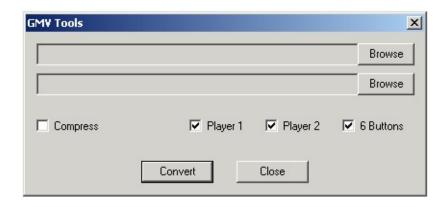
Stop the GMV currently played or recorded

#### **GMV>GMV Tools...**

Open the **GMV** tools window

### Copyright ©2007 Kaneda

# GMV Tools



The GMV tools is a way to convert a GMV file to what you need for your games (into a GMC file).

Using the result file, you can add replay or attractive demo to your games.

If a game is loaded, the input and output files are automaticaly defined, else use the Browse buttons

#### **Compress**

Select if you want the GMC file to be compressed (for size issue)

### Player 1

Include player 1 input?

# Player 2

Include player 2 input?

#### **6 Buttons**

Include players 6 buttons?

#### **Convert**

Convert the GMV file to a compressed or not GMC file with the data selected

#### Close

Quit GMV tools

-----

```
- GMC File Description -
uchar Format
    //what is on the GMC File, add the needed values
    //0x01 \rightarrow player1 input
    //0x02 \rightarrow player2 input
    //0x04 \rightarrow boutons 6 added
    //0x80 \rightarrow compressed
uchar reserved[0x0E]
    //futur use
uchar data
if compressed
    the first char is the number of frames of the input
    the second char is the first input (if needed)
    the third char is the second input (if needed)
    the 4th char is the third input (if needed)
else
    the first char is the first input (if needed)
    the second char is the second input (if needed)
    the thirs char is the third input (if needed)
```

### Copyright ©2007 Kaneda

# Links

Here are the links you should check times to times Come on KMod homepage to jump to the forum and gives you feedback/ report bugs

# Copyright ©2007 Kaneda