

# XGngeo documentation

---

Version 15, released on 2005-06-XX.

## The Choplair-network

Copyright 2003,2004,2005 Choplair-network.

Copying and distribution of this file, with or without modification, are permitted in any medium without royalty provided the copyright notice and this notice are preserved.

---

# Table of Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Getting started</b>	<b>2</b>
2.1	Prerequisites	2
2.1.1	Emulator and Bios	2
2.1.2	Other requirements	2
2.2	Getting and launching XGngeo	2
2.3	First time important path configuration	3
2.4	Main window	3
2.4.1	Menu bar	3
<b>3</b>	<b>Details</b>	<b>4</b>
3.1	Used configuration files	4
3.1.1	Main configuration files	4
3.1.1.1	‘gngeorc’	4
3.1.1.2	‘xgngeo.conf’	4
3.1.2	Rom-specific configuration files	4
3.1.3	Rom driver file	4
3.1.4	History file	4
3.2	Particular windows	4
3.2.1	Rom list	4
3.2.1.1	Preview images (add-on)	5
3.2.1.2	Rom informations (add-on)	5
3.2.2	Global/Rom-specific emulation configuration panels	5
3.2.2.1	Keyboard panel	5
3.3	Internationalization	5
3.3.1	Making new translation	5
<b>4</b>	<b>Web links and ML</b>	<b>7</b>
4.1	Home pages	7
4.1.1	The great duo	7
4.1.2	Their dependencies	7
4.1.3	Other	7
4.2	Gngeo mailling list	7
<b>5</b>	<b>Credits</b>	<b>8</b>

# 1 Introduction

This is the official documentation for **XGngeo**, a frontend (graphical user interface) for **Gngeo**, a command line **Neo Geo** (arcade game playing system made by SNK) emulator for the *GNU/Linux* operating system (and may be some other Unices) with high speed performance and many configurable options. On top of all, both are free/libre softwares released under the terms of the *GNU General Public License*.

Gngeo is written in *Python* and uses the *PyGTK* library to provide a practical and user-friendly *GTK+* interface! This program development is conducted by the **Choplair-network** crew, the latest version and informations about XGngeo are available from our website (see chapter 4 section 1).

Though we are not directly taking part in the Gngeo development (and are not the official frontend too), we follow it closely and try to implement all its functionalities in the most correct way possible, in accordance with the author (Mathieu Peponas) who gently encourages interactions between frontends and its emulator.

This paper, which is supposed to grow up with each new XGngeo release, provides a sort of *Newbie Guide* to get started with easy Neo Geo emulation combining these two programs, and also some details about this frontend's features and its functioning.

## 2 Getting started

This is the complete procedure about how to get the dynamic combination of Gngeo and XGngeo fully functional from scratch, step by step!

### 2.1 Prerequisites

#### 2.1.1 Emulator and Bios

XGngeo is a frontend for Gngeo, so this one must be installed somewhere on your computer! You can download the last version from its home page (see chapter 4 section 1). Although the emulator is available in various binary formats (deb, RPM...), keep in mind that these packages may be older than the classic source code archive. Anyway, consider that any 0.6.x version or more of Gngeo is fine for being used under XGngeo.

Before installing Gngeo, you have to check for the some dependencies: Gngeo graphical rendering is based over the **SDL** (*Simple Directmedia Layer*) library (version 1.2 or more), which is thus required, with **OpenGL** headers for its OpenGL blitter. It also needs **zlib** to extract Rom zip archives. Optionally, you may install the **NASM** package (version 0.98 or more) in order to provide assembler optimisations. Their home page adress, for information purpose, are given on chapter 4 section 1.

Once this is okay, you can perform the installation of Gngeo, in the way corresponding to the format it was grabed into. If, for some reasons, the installation would not succed, you may ask for help to the Gngeo mailling list (chapter 4 section 2).

To be able to launch Roms, Gngeo will need a Neo Geo Bios. It is an archive which consists of the following files: `'neo-geo.rom'`, `'ng-sfix.rom'` and `'ng-lo.rom'`. They have to be put into the same directory as your Roms (the so called *Rom and Bios directory*!).

#### 2.1.2 Other requirements

Especially for XGngeo, you also need the following softwares to be installed:

- The **Python** programming language: version 2.2 or more.
- The **Gimp Tool Kit** aka GTK+: version 2.6 or more.
- The **PyGTK** library: version 2.6 or more.

That's perhaps already the case. Otherwise, you'll find the links to their home page on chapter 4 section 1.

## 2.2 Getting and launching XGngeo

It's show time! Your are now ready to taste the power of XGngeo. ^^

If you didn't obtain this documentation as a part of a XGngeo package, you have to get one! You'll find download links for the last version on the Choplair-network homepage (see chapter 4 section 1). Thereafter, unpack the archive, then move to the directory which have just been created (something like `'xgngeo-XX'` where *XX* is the version number).

**At this time, we *do* assume that you have installed everything indicated above.** Since Python code is interpreted, XGngeo doesn't require any compilation phase. To launch it, simply enter: `python xgngeo.py`. If you get an error doing so, please refer to the chapter 4 section 2.

## 2.3 First time important path configuration

At the beginning, XGngeo should invite you to set up some important parameters that are required to build up basic configuration files with two important options required for a working emulation. The first one, the path to your '`romrc`', is set with the default value. Actually, you would have just to tell the directory where are located your Neo Geo Roms and Bios.

Once you have finished, press `SAVE`. Of course, these parameters can be modified at any time thereafter, using the same configuration window.

If for some reason you want to pass out this important path check at boot time, it is possible by just giving `--nobootcheck` as a command line parameter. But this is definitively **not** recommended!

## 2.4 Main window

Unless you obtained a warning dialog because some parameters looked invalid, the XGngeo main window should appear. At that time, you may fall in love with the marvellous logo, but, actually, your attention should go to the menu bar, on the upper part, which you'll be able to master everything with!

### 2.4.1 Menu bar

It is formerly composed of the `FILE` menu, which permit you do simple operations such as loading Rom (from the Rom list, a file chooser, or a recent Rom history), starting or stopping it, and exiting the program.

Next to that is, comes the `CONFIGURATION` menu, from where you can modify the parameters which you entered at the XGngeo first load and also some other little options, mostly related to XGngeo's own behaviour. But the most interesting here is the `GLOBAL EMULATION` sub-menu which allow you to set the default Gngeo emulation configuration (graphic, audio, keys, etc.) for every game (you can also set specific game configuration). These emulation configuration panels are detailed in the section 2 of the next chapter.

At last, on the very right-hand, from the `INFO` menu you can look at the credits or read again an again the holy GNU General Public Licence which XGngeo is released under!

## 3 Details

### 3.1 Used configuration files

#### 3.1.1 Main configuration files

XGngeo's configuration interface actually manages options of 2 main configuration files at the same time. Both using the same syntax, which is just lines of a variable name followed after a space by its corresponding value. :p

##### 3.1.1.1 'gngeorc'

This is the Gngeo's global configuration file, situated in the '`~/gngeo/`' directory. It lets you customize many params of the emulator, whom some are highly important (path to the '`romrc`' file, etc.), which will be the default for any Rom.

If at startup XGngeo doesn't find the '`gngeorc`' file, it invites you to create it by asking you some required parameters (see above).

##### 3.1.1.2 'xgngeo.conf'

This is the XGngeo's own configuration file, situated in the '`data/`' directory which is in the XGngeo's. This second file is less important, there are only options related to XGngeo (size of history, preview images' directory, etc.). That's why most of these options are modifiable in the OTHER THING section.

### 3.1.2 Rom-specific configuration files

Since its version 0.6, Gngeo is able to perform emulation in a specific way for each Rom.

That's quite simple: before loading the Rom, the emulator looks for a file, in the '`~/gngeo/`' directory, which is named in the form of '`name_name.cf`' (where *name\_name* is the Mame name of the game). If it does exist, the emulation parameters from are taken from, without taking care of the ones set in the '`gngeorc`', which is used otherwise. The syntax for these files is still the same.

Rom-specific configuration files can be easily handled through XGngeo, as detailed in the next section...

#### 3.1.3 Rom driver file

#### 3.1.4 History file

### 3.2 Particular windows

#### 3.2.1 Rom list

There are also optional features (add-on) which bring you a more comfortable game selection in that Rom list window.

### 3.2.1.1 Preview images (add-on)

XGngeo is able to display a preview image of any of the games selected in the list. It is fully compatible with the preview images used by other frontends such as **GGF** (*GnGeo Frontend*), which implemented it formerly.

Thus, a preview image pack archive can be easily obtained from the Gngeo, GGF or Choplair-network home page. You will need to unpack them somewhere, then to indicate the directory where they are located in the OTHER THING CONFIGURATION window, in order to get it working instantly!

### 3.2.1.2 Rom informations (add-on)

GGF's developers created an XML file containing informations for each Rom (description, manufacturer, year, etc.). This is the perfect addition to preview images and it's fully supported by XGngeo! Moreover, because of the small size of that file, it is already included in our packages and this option is activated by default.

## 3.2.2 Global/Rom-specific emulation configuration panels

The options which can be set in global or rom-specific emulation configuration window are exactly the same. As XGngeo provides graphical management for a lot of Gngeo parameters, they have been divided in several panel according to which emulation domain they are dealing with, for the sake of clarity.

### 3.2.2.1 Keyboard panel

XGngeo has a keyboard configurator which permits you to easily customize the 2 player controls. To modify a key, just click on the corresponding button then push your new key.

**Warning:** since Gngeo (SDL's) and XGngeo (GTK's) keymaps are different, some special keys might be not recognized by XGngeo. If it occurs, please tell us (refer to the chapter 2 section 2)!

## 3.3 Internationalization

XGngeo is multilingual! Translations are currently available in the following languages:

- English (default)
- French
- Polish\*
- Portuguese of Brazil\*
- Spanish

**Note:** the translations of a language followed by an asterisk, are, unfortunately, not up-to-date with the current release original strings. Don't hesitate to update them! You can even make new translation, just look bellow...

### 3.3.1 Making new translation

If you want to perform a new translation of XGngeo into your language, follow these generic instructions:

1. Get the current XGngeo development version from its CVS by doing the following command (on a single line):  

```
cvs -z3 -d:pserver:anonymous@cvs.xgngeo.berlios.de:/cvsroot/xgngeo  
checkout xgngeo
```
2. Go to the ‘`lang/`’ directory which is in ‘`xgngeo/data/`’.
3. Create a directory named like your language code (usually the one returned by `echo $LANG`). See [http://www.gnu.org/software/gettext/manual/html\\_chapter/gettext\\_15.html#SEC221](http://www.gnu.org/software/gettext/manual/html_chapter/gettext_15.html#SEC221) for an exhaustive list.
4. Into this one, create a new directory named ‘`LC_MESSAGES`’.
5. Go back to the ‘`lang/`’ directory then open the translation template ‘`xgngeo.pot`’ with your favorite translation tool (KBabel, GTranslator, Poedit, etc.).
6. Once you’ve finished translating. Save it as ‘`XX/LC_MESSAGES/xgngeo.po`’ (where `XX` is your language code).
7. Go to ‘`XX/LC_MESSAGES/`’ then do: `msgfmt xgngeo.po -o xgngeo.mo`. This will create a binary file (‘`xgngeo.mo`’) readable by the program.
8. Launch XGngeo, which is now translated in your language!

Every time you update your translation, don’t forget to repeat step number 7. When you feel it’s perfect, you may send your *po* file to [chopinou\[AT\]choplaire.org](mailto:chopinou@choplaire.org).



## 4 Web links and ML

### 4.1 Home pages

#### 4.1.1 The great duo

- Gngeo homepage: <http://m.peponas.free.fr/gngeo/>.
- Choplair-network home page (for XGngeo): <http://www.choplair.org/>.

#### 4.1.2 Their dependencies

- SDL: <http://www.libsdl.org/>.
- Zlib: <http://www.zlib.org/>.
- NASM (optional): <http://nasm.2y.net/>.
- Python: <http://www.python.org/>.
- GTK+: <http://www.gtk.org/>.
- PyGTK: <http://www.pygtk.org/>.

#### 4.1.3 Other

- Gngeo Brazil (Brazilian website about Gngeo): <http://www.gngeo.hpg.ig.com.br/>.
- GGF (previous frontend) homepage: <http://gngeofrontend.sourceforge.net/>.

### 4.2 Gngeo mailing list

If you get any problem using Gngeo, directly or through a frontend such as XGngeo, the best way is certainly to ask for help on the official Gngeo mailing list, where you should get quick and effective answers from its little community, including XGngeo developers.

Here is how to:

- **Suscribe:** send a blank email, with the word *subscribe* as a subject, to [gngeo-request@ml.free.fr](mailto:gngeo-request@ml.free.fr). Note that prior subscription is mandatory to post any new message.
- **Post:** send your messages to [gngeo@ml.free.fr](mailto:gngeo@ml.free.fr).

By the way, list archives are available on the web at the following address: <http://www.mail-archive.com/gngeo@ml.free.fr/>. You would have better to check out that your problem hasn't been already discussed and solved before posting a new help message.

## 5 Credits

XGngeo forms a part of the projects conducted by the Choplair-network, an independant libre software development crew. Here comes the people involved in the making of this program since the beginning:

- **Choplair** ([chopinou\[AT\]choplair.org](mailto:chopinou@choplair.org)): development director and main programmer, French translator.
- **Pachilor** ([pachilor\[AT\]choplair.org](mailto:pachilor@choplair.org)): assistant programmer.
- **Shilon** ([sheng.long.gradilla\[AT\]gmail.com](mailto:sheng.long.gradilla@gmail.com)): Spanish translator.
- **Matma** ([matma\[AT\]irc.pl](mailto:matma@irc.pl)): Polish translator.
- **Matheus Villela** ([matheusvillela\[AT\]list.ru](mailto:matheusvillela@list.ru)): Brazilian translator.
- **Paulo Eduardo Chiva** ([paulo.chiva\[AT\]ig.com.br](mailto:paulo.chiva@ig.com.br)): previous Brazilian translator.
- **Ms. Marie-Claire** ([marie-claire\[AT\]choplair.org](mailto:marie-claire@choplair.org)): documentation editor.
- **Hori** ([marie-claire\[AT\]choplair.org](mailto:marie-claire@choplair.org)): English spelling corrector.

Special thanks to the Gngeo author, **Mathieu Peponas**, for writing such a great emulator!