

GMenu2X

for the



An Illustrated User's Manual

GMenu2X is an alternative frontend for the gp2x that has been ported to the Wiz and Dingoo A320.

GMenu2X's main feature is the ability to create direct links to your games & applications so that you can directly launch them without having to browse through your entire filesystem manually to run the program every time.

Other features include: built-in selector browser for emulators, manuals and readmes integration, built-in overclocker, gamma adjustment, volume setting, and a ram timings tweaker.

GMenu2X is totally skinnable and there's no risk of bricking your Wiz by installing it.

I started using Gmenu2x on my Dingoo about 8 or 9 months ago due to Dmenu exacerbating a file corruption bug in Dingux. Dmenu was basically a copy of the Dingoo native menu, which in turn was a copy of the playstation/PSP menu system. Dmenu also required you to manually type in any changes you wished to make using a text editor, (you couldn't make changes if you didn't have a computer with you.) Gmenu2x changed all that; additions, deletions, change icons, read text manuals on your handheld, and see 20 icons on screen with scroll capability to view more, all without the need for a computer.

I purchased my Wiz about three months ago, the first thing I noticed was the default menu sucked badly. It was totally useless if you had more than a couple of programs on your wiz. Thank goodness gmenu2x had been ported to the Wiz. GPH seems to be more interested in the design aesthetics of the handhelds themselves than in the user interface. Case in point, look at the Wiz. GPH would have sent the wiz to market with a d-pad on the right side that would only allow you to press one button at a time instead of proper A, B, X, Y buttons if the user base had not pleaded with them to rethink their design. They liked the look of two d-pads. More recently, look at the Caanoo. GPH finally put proper A, B, X, Y buttons on it but, they put a joystick instead of a proper d-pad. It looks great but in my opinion, just about useless for old-school emulation.

I have noticed over the past few months of reading through the Dingoo and Wiz forums that a lot of users (no technical ability I suspect), seem to have problems with gmenu2x, especially on the Wiz. That was the motivation for me to create this guide. One of the main problems users are having with gmenu2x on the Wiz, is when they exit a program that they launched from gmenu2x, the wiz's original menu comes up instead of the gmenu2x one. This can be frustrating and then they blame gmenu2x for the problem and never use it again.

Well it turns out the fault actually lies with the Wiz. The wiz does not re-launch its menu when you terminate a program. You actually have to launch the menu yourself or else you will be greeted with a black screen when you terminate a program. Most programmers do this by launching their program through a bash script. For example the bash script to launch scummvm is called scummvm.gpe and it is shown on the next page;

```
#!/bin/sh

# Export the location of any libs ScummVM depends on
# (to avoid installing to the NAND and overwriting the broken ones there).
export LD_LIBRARY_PATH=`pwd`/lib:$LD_LIBRARY_PATH

# Run ScummVM, important this bit.
./scummvm.gph --fullscreen --gfx-mode=1x

# Sync the SD card to check that everything is written.
sync

# Return to the GPH menu screen
cd /usr/gp2x
exec /usr/gp2x/gp2xmenu
```

The last two lines of the program are run when the program terminates. The first line changes the directory to the location of the original menu and the second line executes the gp2xmenu which runs the original menu. This will start the original menu instead of gmenu2x on program termination whether you have set wrapper=on or not. If the program you wish to add to gmenu2x uses a bash script to run the executable, then it is quite simple for this to work with gmenu2x. To make sure the original menu does not run, the solution is to open the .gpe file in a text editor that allows for Linux text files (DO NOT USE Windows notepad, I use notepad++), and delete the two offending lines:

```
cd /usr/gp2x
exec /usr/gp2x/gp2xmenu
```

Unfortunately, some programmers/porters have discovered that they do not have to write a bash script to start their programs if they put those two lines in their source code and compile their program. This works great if you use the wiz's original menu, but it won't work for gmenu2x. The solution to this problem is a little more complex. You need the source code for the offending program so you can search it for the two offending lines above. Delete these lines from the source code and recompile the program (I did this for all the emulators and games zx81 released and also for MAME). But, if you don't know how to program, this won't work for you. The real solution is for developers to stop compiling these two lines into their programs and learn how to write a simple bash script. That way no matter which menu you like to use, the program will work and call the appropriate menu when it terminates.

Installation:

Copy the file autorun.gpu and the gmenu2x directory to the root of your SD card. Check that the autorun option is enabled on your gp2x wiz. Create the links and enjoy your finally enjoy your gp2x wiz.

Controls:



Main Display:

- L** Moves the section cursor left and displays the icons for the programs in that section.
- R** Moves the section cursor right and displays the icons for the programs in that section.
- D-Pad (L, R, U, D)** Moves the selection cursor the corresponding direction through the sections icons.
- B** Launch the selected program or confirm an action.
- Y** Displays the manual or readme for the highlighted program (*if it has one*).
- Volume up** Increase the CPU clock for GMenu2x program only.
- Volume down** Decrease the CPU clock for GMenu2x program only.
- Volume up + down** Reset the CPU clock to the default value.
- Select** Bring up the contextual menu, (*add link, edit link, delete link, add section, rename section, delete section, and scan for apps and games*).
- Start** Displays the options menu for the GMenu2x program.



File Browser:

X or D-Pad Left

Left Shoulder

Right Shoulder

B-Button

Previous directory.

Page up (when in a list).

Page down (when in a list).

Enter a folder, select a file or confirm an action.



Text Entry Screen:

D-Pad (L, R, U, D)

Left Shoulder, X

Right Shoulder

Y-Button

Moves the highlight area around the virtual keyboard.

Backspace, (erase the character to the left of the cursor).

Insert a space in the text area.

Cycle through the 7 different virtual keyboards, (*Small English letters, capital English and punctuation, symbols and punctuation, small accented letters, capital accented letters, small Cyrillic, and capital Cyrillic*).

B-Button

Selects the highlighted key.

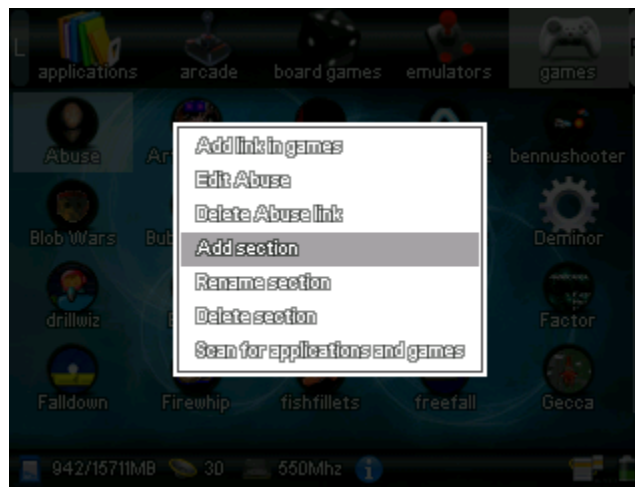
Add a new section

GMenu2x comes pre-configured with four sections:

- Applications
- Emulators
- Games
- Settings

You can add more sections to the sections icon bar at the top if you want to. There are two methods available to add a new section. The first way is to use the context menu:

Context menu



1. Press the Select button on your wiz; this will bring up the context menu.
2. Press down on the wiz's d-pad to move the highlight bar down to the Add section.
3. Press the B button on the wiz to select the Add Section.

This will bring up the text entry screen and prompt you to type the name of the new section that you wish to add.

4. After you have entered the name for your new section, highlight the OK button on the virtual keyboard and press the B button to confirm the name of the section.

The new section that you just added will now appear in the sections menu. If you already have an icon in the "gmenu2x\skins\ (whatever skin you have selected in the gmenu2x settings)\sections" directory and it has the exact name of the new section you just created, then that icon will be in the sections menu. Otherwise, a generic folder icon will be used by default.

The second way to create a new section is to manually create it using a text editor.

Note: Linux and Windows use different ways to denote their end-of-line. If you are using a Windows machine to edit text files for a Linux device such as your Wiz, you must use a text editor that can work with Linux text files. I use Notepad++, but there are many others available just do a Google search.

You can create as many sections as you want. Just create a new directory under sections/ and add a corresponding icon with the exact same name as the directory plus the '.png' extension.

For example if you want to create an "emulators" section, you have to create the directory 'gmenu2x/sections/emulators' and create an icon in the directory 'gmenu2x/skins/ (name of skin directory you are using)/sections' and name it "emulators.png".

Add a new link in a section

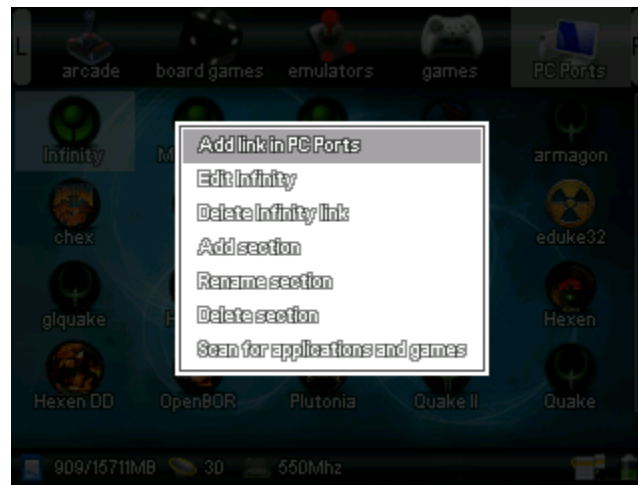
There are two methods to add a new link in a section as well, using gmenu2x and manual using a text editor.

Add a link using gmenu2x:



1. Press the left or right shoulder button on your wiz to move the highlight cursor left or right through the sections bar until it is over the section that you want to add a new link in. In the above picture I have the PC Ports section highlighted, so this will add a new link in the PC Ports section.

2. Press the select button on your wiz to bring up the context menu.



3. With the 'Add link in PC Ports' option highlighted, press the B button on your wiz. This will bring up the File Browser utility.



4. Browse to the directory where the new emulator that you are adding to the emulators section is located, (nand is the internal storage of the wii and sd is the external SD card).



5. I selected sd, this brings up the directory structure of my SD card.
6. Use the d-pad to move the highlight line to the directory that your new emulator is in.



7. Use the d-pad to move the highlight line to the Doom1.gpe file. Press the B button on your wiz. This will add a link to Doom 1 to the PC Ports section and go back to the PC Ports section icons.

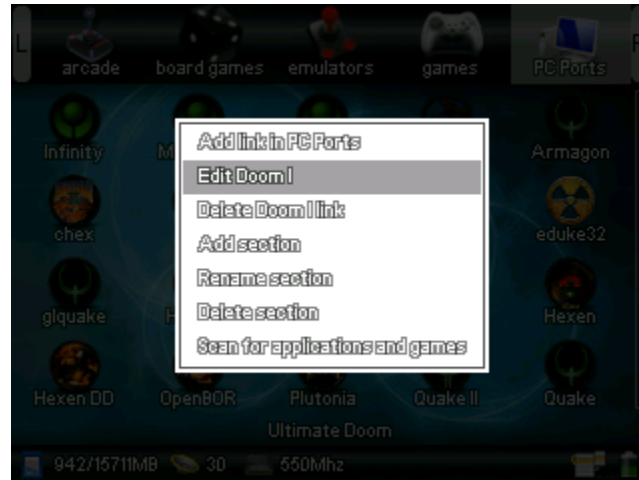


8. Now we have to edit a couple of things in the new Doom I link.



Note: The new link that you just created will appear at the bottom of the icons and will use the default icon, (a white gear).

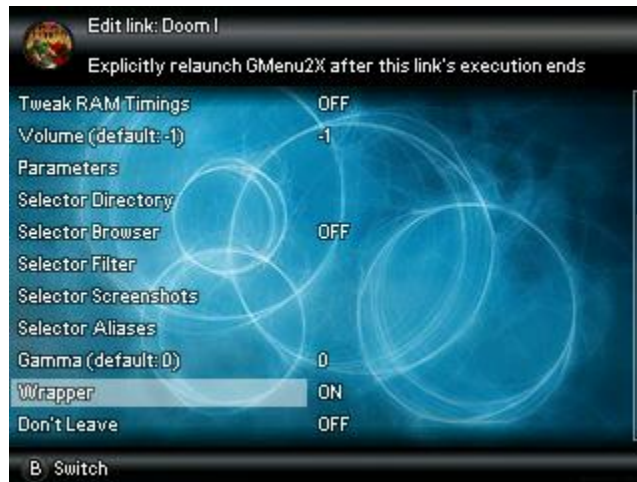
9. Use the d-pad to move the highlight to the new Doom I link. Press the select button on your wiz to bring up the context menu.



10. Use the d-pad to move the highlight line down to the 'Edit Doom I' line and press the B button to bring up the edit link utility.



11. First thing that you want to do is to use the d-pad to move the highlight line up twice to the Wrapper line.



12. Press the B button on your wiz to toggle this switch on.

Note: The wiz does not automatically reload its menu program once a program exits. Setting the wrapper option to 'ON' will wrap the new link in a script that will run gmenu2x when you exit the program, (Doom I in this case). If you forget to do this, your wiz will display a black screen when you exit a program.

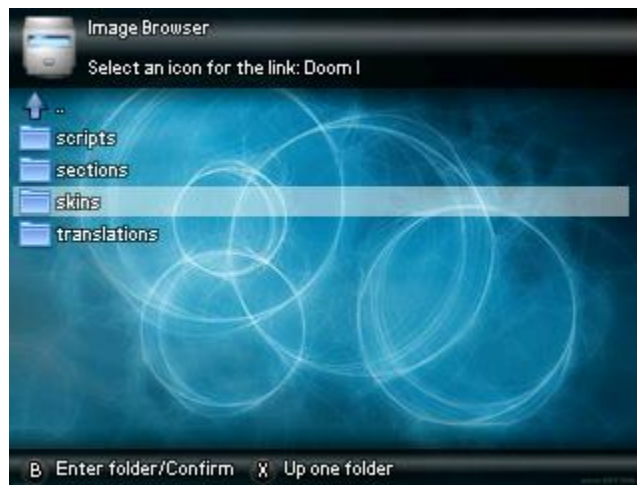
13. Use the d-pad to move the highlight line back to the Title line. Press the B button to enter the title. This will bring up the virtual keyboard utility that will allow you to input the name you want to appear under the icon for the new link. By default, the title will be the name of the executable/script you selected to run the new link. In this case it would be Doom1.gpe. I changed the title to Doom I.



14. Next, use the d-pad to move the highlight line to the icon line. Press the B button on your wii to bring up the image browser utility.



15. Use the d-pad to move the highlight line down to the gmenu2x directory. Press the B button to enter the gmenu2x directory.
16. Use the d-pad to move the highlight line down to the skins directory. Press the B button to enter the skins directory.



17. Use the d-pad to move the highlight line down to the skin directory that you are using. I am using a modified Fast Forward skin (FFW).



18. Use the d-pad to move the highlight line down to the icons directory. Press the B button to enter the icons directory.




19. Use the d-pad to move the highlight line down to the port directory. Press the B button to enter the port directory.



20. Use the d-pad to move the highlight line up or down to the icon you wish to use for the new link you created. Press the B button to select the icon you chose.



The two screen shots above show all of the options available when you select edit a link. I will list each option and a description of what it does and how to use it below.

- **Title** - This is the title of the application/game, its value is only used for display purposes.
- **Description** - A short description of the application/game. This will be displayed centered on the bottom of the sections icon screen when the icon is selected, (Useful when the game name is too long to display under the icon).
- **Sections** – This lists what section the icon will be displayed in.
- **Icon** - The path to the icon for the application/game. If a path is not specified, a default icon will be used, or if the application/game directory contains a png icon file with the exact same name as the executable file but with a .png extension, that will be used. The icon size should be 32x32px, (*optional*).
- **Manual** – Path to the manual or readme text file for the game/emulator. If a manual is available for the selected icon a small icon  will be displayed in the bottom bar of the icon screen. If when you created the link to the game/emulator the directory contained a text file with the same name as the new link but with a .txt extension, gmenu2x will put the path to that file here for you.
- **Clock** - Sets the default clock speed the application/game will use when executed, (can be modified at run-time), (*optional*).
- **Tweak RAM Timings** – This option usually speeds up the application but, at a cost of stability. I have never had the need to try this setting.
- **Volume** - Sets the volume for the application/game will use when executed, (can be modified at run-time). *Default = -1, this means the application/game will use the default volume level your wiz is set at when the application/game is started, (optional).*

- **Parameters** - Optional parameters to pass to the application/game when it is launched.
- **Selector Directory** – The directory displayed when the Selector Browser is ON and the link is executed, *(optional)*.
- **Selector Browser** – Some emulators on the wiz require you to specify which rom you want to run when you run the emulator. When this option is ON, gmenu2x will use its built-in file browser to allow you to select a rom file name to pass to the emulator when it is run, *(optional)*.
- **Selector Filter** - Type the dot three letter extension/extensions that you want to be visible in the Selector Browser. Separate multiple dot three letter extensions with a comma. For example if you only want file names with a '.zip' or '.smc' file extension to be visible on the selector screen (all other files that do not have these dot three letter extensions will not be displayed), then you would type; ".zip,.smc", without the quotes. *(Optional, entries here are only used when the selector browser is used)*.
- **Selector Screenshots** - The path to the directory containing screenshot files in the png format, max size 160 x 160px. Screenshots must have the EXACT SAME NAME as the file selected in the selector browser but with a '.png' extension. The screenshot if available will be displayed in the selector browser when the file with the same name is highlighted. . *(Optional, entries here are only used when the selector browser is used)*.
- **Selector Aliases** - Path to a text file that contains a list of aliases for the files in the selector browser, for example ssriders=Sunset Riders, ssriders is the actual file name and the alias Sunset Riders is what will be displayed in the selector browser . *(Optional, entries here are only used when the selector browser is used)*.
- **Gamma** - Gamma correction controls the overall brightness of an image. Images which are not properly corrected can look either bleached out, or too dark. Use **X** to decrease value and **Y** to increase value, *(Default 0)*.
- **Wrapper** - If this option is **ON**, then the game/emulator will be launched in a script that will restart gmenu2x when the game/emulator is terminated, *(Optional, but this option needs to be on for every link, or else you will get a black screen when you terminate the game/emulator you launched)*.
- **Don't Leave** – If this option is **ON**, then gmenu2x will not terminate when you launch the link. You can use this option to launch a link in the background such as a screenshot application, *(Optional)*.

You can also manually add links to a section by creating a text file and adding it to the desired section in the gmenu2x directory, (i.e. gmenu2x/sections/ *(The name of the section to add the link in)* / *(text file named the same as the linked executable without any extension)*).

A sample link text file looks like this:

```
title=mame
icon=skin:icons/emulators/Arcade.png
exec=/mnt/sd/Emulators/Mame/mame.gpe
manual=/mnt/sd/Emulators/Mame/readme.txt
clock=550
wrapper=true
```

There are a couple of options that are not available when using gmenu2x to add/edit a link that are available when manually creating a link with a text editor, (this is because gmenu2x automatically fills in this information when it creates a link).

- **exec** - this is the path to the file to run when this link is selected. If the file doesn't exist the link will be disabled, so that you can create links for applications that are not installed or that are on different SD cards, and gmenu2x will only show the valid ones.
- **workdir** – If a path is specified, gmenu2x will execute a change directory (chdir) to the specified path before launching this link, (*Optional*). If not specified GMenu2X will chdir to the path of the exec key (i.e.: /mnt/sd/drmd if exec=/mnt/sd/drmd/drmd.gpe).