

Garshneblanker

COLLABORATORS

	TITLE : Garshneblanker		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		October 20, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Garshneblanker	1
1.1	Garshneblanker_Documentation	1
1.2	introduction	1
1.3	copyright and disclaimer	1
1.4	requirements and installation	2
1.5	using garshneblanker	2
1.6	module usage notes	3
1.7	module creation	4
1.8	credits	5
1.9	plans	5
1.10	history	5
1.11	Open... Menu Item	7
1.12	Save as... Menu Item	7
1.13	Load module... Menu Item	8
1.14	About Menu Item	8
1.15	Quit Menu Item	8
1.16	Last Saved Menu Item	8
1.17	Random Preferences Menu Item	8
1.18	Screen Mode Button	9
1.19	Module Preferences Button	9
1.20	Timeout Integer Gadget	9
1.21	Blank Key Text Gadget	9
1.22	Pop Key Text Gadget	10
1.23	Load Button	10
1.24	Hide Button	10
1.25	Save Button	10
1.26	Quit Button	11

Chapter 1

Garshneblanker

1.1 Garshneblanker_Documentation

Garshneblanker, A System Friendly Solution to Blanking Screens

Select from the following topics:

- Introduction
- Copyright and Disclaimer
- Requirements and Installation
- Using Garshneblanker
- Module Usage Notes
- Module Creation
- Credits
- Plans
- History

1.2 introduction

As you may have already noticed, Blanker is no longer Blanker. It is now Garshneblanker. I had received numerous suggestions to change the name from Blanker to something else, so I finally got around to doing it.

Garshneblanker is an attempt to provide the Amiga community with a future compatible, easily expandable screen blanker.

I would like to provide a platform for others to write custom screen blanker modules when the whim strikes and not have to worry about the difficulties associated with setting up a Commodities interface and dealing with concurrency problems.

1.3 copyright and disclaimer

Garshneblanker is Copyright (c) 1993 Michael D. Bayne.
All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgement:
This product includes software developed by Michael D. Bayne.
4. My name may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY MICHAEL D. BAYNE ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL MICHAEL D. BAYNE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Note: Nico François' ReqTools library is included in this distribution, but these terms do not apply to that portion of the distribution. Refer to the documentation in the ReqTools distribution for such information.

1.4 requirements and installation

This software requires that you be running AmigaDOS(tm) 2.04 or higher.

Additionally, reqtools.library (Copyright Nico François) is used by Garshneblanker and is consequently included in the distribution archive.

Installing Garshneblanker is a very simple process. Double click on the installation icon and click on the pretty buttons.

1.5 using garshneblanker

I have layed out here in a hopefully intuitive manner, Garshneblanker's interface so that you can simply click on the gadget or menu item about which you have question.

Project	Edit	Settings	
Open...		Last Saved	Random Prefs?

```

Save as...
Load module...
-----
About...
-----
Quit...

+-----+
| Screen... (Screen mode description_____) (# of Colors)
|
| Prefs... (Description of the currently selected module_____)
+-----+
|          | Timeout   (_____)   Load...   Save
|          |
| (Image)  | Pop Key   (_____)       Hide
|          |
|          | Blank Key (_____)       Quit
+-----+

```

Also, there are two standard tooltypes supported by Garshneblanker. They are:

CX_POPUP: If you set CX_POPUP=YES in the tool types section of the Info panel for the Garshneblanker icon, Garshneblanker will popup its interface when it is run. If it is set to CX_POPUP=NO, Garshneblanker will remain hidden on startup and can be popped up with the Pop key or Commodities exchange.

CX_PRIORITY: This will probably not need to be changed for normal use. It is implemented for completeness. CX_PRIORITY is set to a number that tells the Commodities brokers who to send input events to first. This way if two Commodities are monitoring the same key (the Pop key, for example) the one with the higher priority gets that key-press first.

1.6 module usage notes

1. Text module notes

In an effort to keep the Text module from getting any bigger, I didn't put a file requester in it for selection of a text file from which to pick quotes. So, when you put it in "Files" mode, you just type the full (or at least relative to the directory that Garshneblanker was started from) path name to the text file and Garshneblanker will pick a random quote from that text file.

Also the format is:

```

Number of quotes in file    <-- No extra returns
Font_Name Font_Size Quote  <-- between any lines.
...

```

See the example, Text.quotes.

1.7 module creation

I've tried to make writing your own Garshneblanker module as simple a task as possible. So let me elucidate the few things that you must keep in mind when writing your own module:

Your module must have the following three functions:

```
VOID blank( VOID );
VOID prefs( LONG command );
VOID defaults( struct bMessage *bMsg );
```

I have provided for you a main.c file which you can Link with your blank(), prefs(), and defaults() functions. It sets up a message port and communicates with the Garshneblanker program taking care of all the details.

The best things to do is look at the other blanker sources to find out how a module should be written. However, I'll try to sum up what each function requires and performs in case my code isn't clear enough.

blank() -- This function is provided external variables Depth and Mode and should blank the screen using that screen mode and screen depth if possible. blank() has to do all waiting on SIGBREAKF_CTRL_C as that is the signal it will get when it should stop blanking. Other than that, blank() can also call the BlankMousePointer() and UnblankMousePointer() functions that I have provided if it wants system friendly mouse blanking.

defaults() -- This function simply fills in the 512 byte block of data that contains the default preferences for the module. This 512 byte block of data will be saved in preferences files and sent to blank() and prefs() so they can use it and change it's values respectively. In order to be sure that this 512 byte block of data always has valid values for this module, defaults() will be called to initialize a newly allocated prefs structure. Also, in this function you can specify a screen mode or depth that your module requires and Garshneblanker will disallow the manipulation of that value in it's settings display. Lastly, the module can specify that it has no preferences (other than screen mode and screen depth, I would hope) and Garshneblanker will not attempt to open a preferences panel for your module.

prefs() -- This function is a bit more complicated. Again, I urge you to look at the existing code. This function responds to three commands (which are passed in the 'Command' argument).

START - This command tells prefs() to open it's window and do any other initialization that must be done.

IDCMP - This command tells prefs() that an IDCMP message has arrived at it's window port and that it should handle that message at this time.

KILL - This command tells prefs() to clean everything up and shutdown. Closing its window and anything else it had open.

These commands are #defined in defs.h.

Finally, a small function winSig() must be written that returns the prefs window message port signal when the window is open and null otherwise. Garshneblanker uses this when monitoring for signals so that it knows when

to tell prefs() to handle its IDCMP signals.

1.8 credits

I must first thank Alexander Kneer for sending me the source to some modules he ported from other screen blankers. I used those to create six of the new blankers you'll find in this distribution. Also, for help with AGA bitmaps since I don't have an AGA machine to work on. And, thanks to Tobias Abt for referring me to Alex and the new modules, etc.

Second, let me express my extreme gratitude to Nico François for writing his wonderful ReqTools library. It makes it very easy to have a complete professional look without having to write a great deal of "requester" code. I applaud the quality of the ReqTools package.

Last, I also express gratitude to Jan van den Baard for writing his amazing GadToolsBox interface toolkit.

Author Information:

Michael D. Bayne, CS Undergraduate at Rose-Hulman Institute of Tech.

E-Mail address:

baynemd@nextwork.rose-hulman.edu (bug reports, etc. here)

Mailing address:

Box 216 (mail bombs, etc. here)
Rose-Hulman Institute of Technology
5500 Wabash Ave.
Terre Haute, IN 47803

1.9 plans

Still no AREXX port. Bear with me.

Also, if you have any good ideas for Garshneblanker modules and you don't feel up to the adventurous task of writing them yourself, feel free to drop me a line with the suggestion and I'll see what I can't whip up and include in the next release of Garshneblanker. Of course this goes as well for suggestions and ideas for the actual Garshneblanker program as well.

1.10 history

Version 1.0 - Initial public release of Garshneblanker

Version 1.1 - Fixed bug in Moire module X-Speed gadget
- Main window wasn't 'locked' when pref window was open

(ie. busy pointer, no mouse input), fixed it.

Version 2.0 - Implemented modules and modularity.
- Wrote the Text/Clock module.

Version 2.1 - Replaced CHECKBOX_KIND Resident button with CYCLE_KIND because GadTools.library v39 is so 'bug free' that the checkbox gadget caused enforcer hits and wouldn't render.
- Separated the Text/Clock module into a Text module and a Clock module and added more features to each.

Version 2.2 - Blanked the mouse as well as the screen for people who aren't using a separate mouse blanker.

Version 2.3 - Removed mouse blanking code because system friendly mouse blanking for all versions of the OS is a kludge, and its better for each person to use a standalone mouse blanker (QMouse) that suites their needs.
- Added Copper list options to the existing modules.
- Removed delays before blanking after discovery of 'Upstroke' keyword to commodities HotKey objects.

Version 37.1 - Changed to Commodore's official version/revision style. So, now the version is 37 revision 1 and the release is 2.3.

Version 37.2 - New feature: When modules require a certain screen mode or screen depth, they can just set that mode/depth in their defaults function and Garshneblanker will pick up on this and not allow the user to change the mode or depth accordingly.

Version 37.3 - Must've gotten BumpRev happy, because I don't recall any reason for a revision increment here.

Version 37.4 - New feature: Random preferences mode. You can now set your preferences to load a random file when you blank. See above for more documentation on how to do this.

Version 37.5 - Two new modules: Puzzle and Fade, unfortunately these two modules are untested on AGA machines (my AGA test machine went away for the summer).
New feature: Displays the icon of the current blanker in a nice little window.
Bug fix: I went ahead and changed the input handler to ignore key-ups because the commodities 'upstroke' didn't ignore the 'upstrokes' of the qualifiers, only the main key.

Version 37.6 - Changed preferences files to IFF files in the format C= specifies for preferences files.

Version 37.7 - Rewrote the entire Garshneblanker <--> Module message passing algorithms. The new system is much more asynchronous and (unfortunately) not compatible with older modules. BUT since no one has exactly jumped up and released any Garshneblanker compatible modules, I'm not too concerned.

The new method allows for Blanking with or without module prefs windows up and makes it really easy to test preferences while still in the window.

- Added a 'Test' button to all modules to blank with the temporary settings that are set in the module prefs window.
- Removed the 'Load on request' option for modules. Now, the current module always remains resident and random prefs are moved to a menu toggle. If someone doesn't have the memory for a resident module, they should use QMouse.

Version 37.8 - Added shift-select capability for preferences files. So now you can click on Garshneblanker and a preferences file and it will start up with those pref settings.

- Added AppWindow support to Garshneblanker (this is really easy). So, if you drop a preferences file in Garshneblanker's window, it will load those prefs and use them.

Version 37.9 - Moved a lot of the preferences out of the IFF preferences file and into ToolTypes (thanks to Tobias for this suggestion).

Version 37.10 - Changed random blanker preferences so that the modules is now changed every n seconds where n is the number of seconds the user has set for timeout.

- Wrote a new module: 'Life'.
- Reworked four new modules sent to me by Alexander Kneer (thanks Alex) into six modules and in the process added support for modules with no preferences at all.
- Came up with a reasonable, OS-compatible, mouse-blanking routine and put that in there. I don't like it, but there's not much of a better way currently supported by Commodore.

Version 37.11 - I rewrote just about everything that had to do with the modules and the blanking from an Object-Oriented perspective so it does the same thing for the most part, however is much more sound from a programmer's perspective. Also, the bugs that existed in the last version were addressed (lots of wacky behavior, I apologize for such a poor release).

- I changed the name from Blanker to Garshneblanker.
- Added a new module. Stars. (From A. Kneer as well).
- Added another new module. Fireworks. (Again A. Kneer).

1.11 Open... Menu Item

This allows you to open a preferences file directly. These files are generally stored in the Presets subdirectory of your Preferences directory. A file requester will appear and you can click on the file you wish to load.

1.12 Save as... Menu Item

This allows you to save the current set-up to a file that can later be opened from the Open... menu item. This will bring up a file requester and allow you to enter the name and location of the file. Also you can use this to save preferences with the "Random blanker" mode set in your SYS:Prefs/Presets/Blanker directory so that Garshneblanker will load a random set of preferences when it blanks the screen.

1.13 Load module... Menu Item

Here is where you can load different blanker modules. It has the same function as the Load button.

1.14 About Menu Item

This pops up a requester with Garshneblanker's version number and my e-mail address.

1.15 Quit Menu Item

This is the same as pressing the Quit button in the interface window.

1.16 Last Saved Menu Item

If you make modifications to the current set-up and then decide you would like to start over, this menu item will load the Garshneblanker preferences file again and restore the set-up from your last save.

1.17 Random Preferences Menu Item

Random preferences mode causes Garshneblanker to load a random preferences file from the SYS:Prefs/Presets/Blanker/ directory and use those preferences to blank the screen. With this feature you can have blanker randomly blank from one of a number of random settings whenever the screen blanks. See the example preferences for how you would set this up.

In this mode, Garshneblanker will also load a new module whenever the timeout expires again (so if Timeout is set to 90 seconds, a different preferences file will be loaded every 90 seconds).

1.18 Screen Mode Button

Selecting this button brings up a requester prompting you to select a screen mode. Here you can specify which type of screen you want the blanker to come up on and the number of colors that screen will have. This information is shown in the two text fields to the right.

If this button is 'ghosted' (shaded with black dots and unselectable), then the blanker module that is currently selected does not allow the user to select which screen mode it opens up on. For example, the Puzzle module opens a screen of the same resolution and colors as whichever screen happens to be at front when the screen blanks. Thus any specification of screen mode or colors would be pointless. Also, if the module allows the user to select the screen mode but not the number of colors, the screen depth gadget will not appear in the screen mode requester. The number of colors that the module specifies will be shown in the number of colors field.

1.19 Module Preferences Button

Selecting this brings up the preferences panel for the currently selected blanker module. Adjusting these parameters and pressing the OK button on this panel will change the behavior of the screen blanker module's graphical display. If you press CANCEL on this panel, the changes you made will be ignored.

These preferences will be saved along with Garshneblanker's preferences when you click the "Save" button or select "Save as..." from the menu bar.

This means that module preferences aren't necessarily connected to the module. You can have a number of saved preference files with different combinations of module and module preferences.

1.20 Timeout Integer Gadget

This field allows you to change the length of time that Garshneblanker waits before blanking the screen. This value is in seconds.

This value is stored in the TIMEOUT tooltype which you can also modify.

1.21 Blank Key Text Gadget

This field allows you to change the key that is pressed to blank the screen.

This key is described as a standard Commodities hot key. For more information see page 5-29 of the "Using the System Software" manual supplied with your computer.

This key is stored in the BLANKKEY tooltype which you can also modify.

1.22 Pop Key Text Gadget

This is similar to the blank key gadget. It allows you to set the key that is used to pop up the Garshneblanker window.

Additionally, the Commodities exchange program can be used to pop up the Garshneblanker interface. Refer to page 4-27 of your "Using the System Software" manual for more info on the Commodities exchange program.

This key is stored in the CX_POPUP tooltype as well.

1.23 Load Button

This button pops up a requester in which you can select a different blanker module to use. This new blanker will be loaded and its information will be displayed in the module description box.

If the module cannot be found, the internal (simple black screen) blanker will become active.

Additionally, the module's icon is displayed in the icon well on the left side of the interface.

The directory in which this new module is loaded is now the current Garshneblanker module directory and is will be stored in the BLANKERDIR tooltype if you save the settings.

The name of the module you select is stored in the BLANKER tooltype when you save the settings.

Note: There is only one BLANKERDIR setting and when you use 'Save as' to save a separate settings file, the BLANKERDIR tooltype is not included. Thus you must put all of your blanker modules in the same directory, OR modify the BLANKERDIR tooltype manually in any custom settings files.

1.24 Hide Button

This button hides the blanker interface from view. Garshneblanker is still running when its interface is hidden and you can pop the interface back up with the Pop key or the Commodities exchange program. Also note that the close button in the upper left hand corner of the screen does the same thing as the Hide button. It doesn't actually quit the program.

1.25 Save Button

The save button saves the current preferences so that when you reboot your computer, the settings will be the same. Additionally, the preferences for the selected module are also saved.

1.26 Quit Button

This tells blanker to exit and frees up any resources it was using at the current time.

Additionally, if the Garshneblanker module was resident, it too will be told to exit.
