Blanker

COLLABORATORS

	TITLE : Blanker		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		March 1, 2023	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

Blanker

1.1 Blanker Documentation

Blanker, A System Friendly Solution to Blanking Screens

   Select from the following topics:

   @{ " Introduction   " link introduction }
   @{ " Copyright and Disclaimer   " link copyright }
   @{ " Requirements and Installation " link requirements }
   @{ " Using Blanker   " link usage }
   @{ " Module Usage Notes   " link moduse }
   @{ " Module Creation   " link modules }
   @{ " Credits   " link credits }
   @{ " Plans   " link plans }
   @{ " History   " link history }

1.2 Introduction

Blanker 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

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

Redistribution and use in source and binary forms, with or without modi-
fication, 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 no-
   tice, 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
Blanker and is consequently included in the distribution archive.

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

1.5   Using Blanker

          I have layed out here in a hopefully intuitive manner, Blanker'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 Blanker. They are:

CX_POPUP: If you set CX_POPUP=YES in the tool types section of the Info
  panel for the Blanker icon, Blanker will popup its interface when it is
  run. If it is set to CX_POPUP=NO, Blanker 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 Blanker was started from)
 path name to the text file and Blanker 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 Blanker 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 );
```

The bMessage structure is defined in the @{ "defs.h" LINK "Source/Blankers/defs.h/ ←
   MAIN" } file (which you'll have
to include in your source). Here it is for reference:

```
struct bMessage {
  struct  Message bm_Mess;
  struct  Task *bm_Task;
  ULONG   bm_Mod;
  ULONG   bm_Dep;
  ULONG   bm_Type;
  UBYTE   *bm_Info;
  UBYTE   *bm_Data;
};
```

I have provided for you a @{ "main.c" LINK "Source/Blankers/main.c/MAIN" } file ←
   which you can link with your
blank(), prefs(), and defaults() functions. It sets up a message port and
communicates with the Blanker program taking care of all the details.

The defaults() function simply copies valid default preferences information
into the prefData space. Blanker and @{ "main.c" LINK "Source/Blankers/main.c/MAIN ←
   " } will take care of checking
whether or not the prefs are valid and will call defs() to get valid prefs
when necessary. This guarantees that blank() and prefs() will always be
sent valid preferences information when they are called. Also, if you plan
to force the screen into a particular mode or depth, you can set the sMod
or sDep of the bMessage and Blanker will realize you did this and disable

the mode and depth settings accordingly. Additionally, if you return NULL in prefData, Blanker will assume you have no preferences whatsoever and act accordingly.

The prefs() function will have to open a window, get the preferences settings from the user, copy those into prefData space, close the window, and return. This is all left up to the module for flexibility. The only limitation is that prefData is only 512 bytes, and therefore no more than 512 bytes of information can be stored in prefData. In addition to the prefs function a winSig() function is necessary for the controlling code to wait for signals at your window's port. A quick glance at any one of the existing blankers will show you exactly what to do for this function.

The blank() function will be called and sent the bMessage pointer so that it can extract the additional information of screen depth and screen mode. These are the modes that the user selects in the main Blanker panel. Please try to use this information intelligently and open the closest screen mode that your module supports. These modes will span the range of original/ECS/AGA screen modes so be prepared for this contingency. However, if it doesn't make sense to use more than one bitplane for example, then you have the option to ignore that field. Also, I leave the overscan settings to the module. So that can be set to whatever seems fitting for your module. The blank() function has to be a bit more intelligent than the other functions. It can blank in a repeating loop and should quit when it is sent the SIGBREAKF_CTRL_C signal. The best way to understand how that is done is to simply look at the main loops in the supplied example blanking functions. It is a fairly simple method and works quite well. After the CTRL_C is received, you must close down any opened resources and return.

That's about all the help I think you should need in writing your own blankers. Like I said before, I think the best thing to do is to have a look at the existing blanker modules and fill in your own specific information therein. I will also make the suggestion that using GadToolsBox for your preferences interface allows you to very easily create a preferences panel.

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 Blanker 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 Blanker. Of course this goes as well for
suggestions and ideas for the actual Blanker program as well.

1.10 History

Version 1.0 - Initial public release of Blanker

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 Blanker 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 Blanker <--> 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 Blanker
    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 Blanker and a preferences file and it
    will start up with those pref settings.
    – Added AppWindow support to Blanker (this is really easy).
    So, if you drop a preferences file in Blanker'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 suggest-
    ion).

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.
```

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 Blanker 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 Blanker'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 Blanker
preferences file again and restore the set-up from your last save.

1.17   Random Preferences Menu Item

Random preferences mode causes Blanker 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.

Also, a new random module will be loaded after the 'Timeout' setting
has elapsed.

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 Blanker'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 Blanker 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 Commodites 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 Blanker window.

Additionally, the Commodities exchange program can be used to pop up
the Blanker 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
Blanker 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. Blanker 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 Blanker module was resident, it too will be told to exit.