bgui

COLLABORATORS

	TITLE : bgui		
ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY		April 15, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

bgui

1.1 bgui.guide

```
                BGUI release 1.2
***************


General information:


            Preface
                READ THIS!

            Introduction
                What is BGUI?

            Features
                BGUI features.

            Requirements
                What do you need.

            Installation
                Where to put what.

Programmers information:


            Read Me
                Things you should know.

Users information:


            Keyboard & Mouse
                Keyboard controlled object activation.

Required reading :)


            Disclaimer
```

                    You want Warranty? Buy a toaster... ;)

               License
                    License agreement.

Program information:

               Future
                    TODO.

               Thanks To
                    Thank you very much.

Like it say's... Index:

               Index
                    What's where.

1.2   bgui.guide/Preface

Preface
****

               BGUI release 1.2
               (C) Copyright 1993-1995 Jaba Development
               (C) Copyright 1993-1995 Jan van den Baard
               Written using DICE v3.0 by

          SNail: Jan van den Baard
           Bakkerstraat 176
          3082 HE Rotterdam
               Holland

     Fido: 2:286/407.53 (Jan van.den.Baard)
          EMail: jaba@grafix.wlink.nl

        EMail is preferred.
     If you must use snail mail then please make it clear
     when you absolutely need an anwser and (don't forget
      this) include enough return postage.

          User Manual
     (C)Copyright 1993-1995 Jaba Development
          __ __
         / / / \
      __   / / / /\ \
       _____\ __/ /_/ /_/ /_____
          (_____)

   BGUI, (C) Copyright 1993-1995 Jaba Development. All rights reserved. This
program is GiftWare. This means that if you want to use the programs you are

morally obligated to reward the author with a gift that, you think, matches
his efforts. If you cannot think of a gift (come on! be creative ;), a small
financial donation will always be gladly accepted.

   I have thought about how to release this package for quite some time. My
initial thought went out to shareware but I soon forgot about this. Since most
of the libraries of this kind seem to be shareware I do not think that many
people like this simply because it's anoying to have to pay for a library if
you had to pay for the application as wel.

   To avoid these problems I have decided to release this package as giftware
for people writing freely distributable software. With "freely distributable
software" I mean non-crippeled, free of charge software. Authors who wish to
use BGUI in shareware and/or commercial programs must pay for the usage of
BGUI before actually using it.

   ShareWare and/or commercial usage is not allowed without:

      1) Prior written permission from the author.
      2) A small finacial donation of US$ 20,-, DM 30,- or HFL 30,-.
      3) A free and registered copy of the product including free updates that
         still make use of the library.

   The idea behind this distribution policy is simple. You make money with the
aid of BGUI? I want to get my (very humble) share.

   This software is not crippeled so when you register it you will not recieve
a key-file or an updated version. Registering simply buys you the right to use
this software in shareware and/or commercial applications.

   Resonable, eh?

1.3 bgui.guide/Introduction

Introduction
****

   BGUI is a BOOPSI based shared library contaning a number of BOOPSI classes.
It allows for easy creation and managing of font sensitive and size adjustable
GUI's on the Amiga. Hmm, where have I read something like this before..., I
hear you say.

   YES! This is yet another GUI library. I know there has been a lot of
discussions about the usabillity and neccesity of GUI libraries on the nets
lately. Still, being the ignorant bugger that I am, at some point I felt it
was necessary to write yet another one.

   Believe me when I say that BGUI was not intended to become a releasable
project. I started coding BGUI simply because I was amazed what MUI did using
BOOPSI. Therefore I started coding some BOOPSI classes as a learning
experience.

   Hmmm... This is beginning to sound like an excuse... As if I did something
terrible ;)

The point is that, while working on BGUI, I saw many GUI libraries being released and not one of them really seemed to take off. This made me think that all this talk about "For every app I run I need to install a different GUI library" really is bull. Please correct me if I'm wrong but I have yet to see a program requiring a GUI library (with the exception of MUI ofcourse).

Sure there are a lot available but MUI is the only one which actually seems to be used by the programmers.

This is the reason for BGUI. I have tryed to create a library which does it's thing simular to MUI eliminating the biggest problems that programmers seem to have with MUI. Meaning no fancy background-pen-frame-text-image-whatever settings editor, not a zillion extra images, no external class libraries etc.

BGUI is a single library which can be removed as easely as it can be installed. It provides an easy way to create and control GUI's without becoming excessive in what it does.

I guess time will tell if BGUI is going to be accepted or not...

1.4 bgui.guide/Features

```
Features
********
```

   * Requires at least OS 2.04 (Yes, that is a feature :)

   * Extends on the existing system gadgetclass and rootclass.

   * Creates completely font sensitive GUI's.

   * Supports sizable windows.

   * AppWindow support.

   * Titled group-frames and seperators.

   * Supports OS 3.0 when available.

   * Style Guide compliant.

   * Keyboard control of most gadget classes.

   * Several types of notification to create "self driven" GUI's.

   * Easy intergration of third-party classes.

   * Easy to make your class work directly with BGUI.

   * Object oriented.

   * Resonable speed, even with a 68000 processor.

* Relatively small library.

* Classes for easy filerequester and commodity creation.

* Seperate ARexx class including source code.

1.5 bgui.guide/Requirements

Requirements
****

   BGUI requires at least version 2.04 of the operating system. It does not
run on 1.3 systems. Are there still people running 1.3?

   BGUI also requires the following disk-based libraries to operate:

* gadtools.library   (V37) (menus)

* asl.library   (V37) (filereq class)

* commodities.library (V37) (commodities class)

   The following libraries are not required but they will impose on the
functionality of the library when not available:

* locale.library   (V38) (localized library strings)

* amigaguide.library (V37) (On-line help)

* workbench.library   (V37)   (App-window support)

   It should not be a problem to use BGUI on a system with 512KByte but a
minimum of 1 Megabyte is suggested as a minimum.

1.6 bgui.guide/Installation

Installation
****

   The installation of BGUI is very simple. You can either use the Installer
utility by C= which will install the library for you or you can copy the
bgui.library to your LIBS: directory or into a directory you might have setup
for non-system libraries.

   Where you put the headers/modules for the language you use is up to you.
Since the different compilers and users use different directory structures I
did not supply a script for you to install these. You'll have to do this by
hand.

1.7 bgui.guide/Read Me

```
Read Me
*******
```

  * The bgui.library is a special library in the sence that it is a single
    task library. This means that every task using the library must open the
    library for itself. The library base pointer may not be shared between
    different tasks.

    The reason for this is that the library tracks information like window
    bounds etc. This tracking is done for each individual task.

    So if your code spawns a sub-task which needs the library, the sub-task
    code must open the library for itself. It may not share it with the main
    task.

  * The object ID's (gadgets and menus) are restricted to the range 1..65535.
    ID's smaller or larger than this are not allowed. This should not be a
    problem :)

  * When designing custom classes to work with BGUI please try to keep your
    tag values and method ID's well out of the range of the BGUI tag values
    and method ID's. Using values that conflict with the BGUI values may have
    unexpected results.

  * Do not be alarmed when you notice disk-activity when a BGUI requester
    pops up. This is due to the fact that the bgui.library catalog file is
    re-loaded each time BGUI needs it. This makes sure that the language is
    correct if it changed.

  * When scrolling a large listview the mouse pointer may move in steps
    rather than smoothly. This is not abnormal behaviour with BOOPSI objects
    because all rendering occures on the input device it's task. Just scroll
    a MultiView window with the scroller and you'll see the same effect.

  * The BGUI windowclass depends on the IDCMP_SIZEVERIFY flag. This means
    that the responce to the signals you get have to be as quick as possible.
    If you need to use DOS routines or are unable to call upon the
    windowclass event handler for a while you need to switch of the
    IDCMP_SIZEVERIFY dependance before you do and back on after you're done.
    There is a tag called WINDOW_NoVerify for this.

1.8 bgui.guide/Keyboard & Mouse

```
Keyboard & Mouse
****************
```

   Most of the BGUI gadget objects can be controlled by the keyboard. Usually
the key that is used to control an object is shown by an underlined character
in the object it's label. The objects can be controlled as follows:

  *   Button - Pressing the key will make the object appear pressed. If you

release the key the object appears normal again and the application is
signalled that the object was selected. When you accidentally activated
the button you can press SHIFT or ESC before releasing the key to cancel
the selection.

*   Toggle Button – Pressing the key will toggle the selected state of
the toggle button.

*   CheckBox – Pressing the key will toggle the selected state of the
checkbox object.

*   Prop (Scroller) – Pressing the key will advance the knob position by
one page. When you press the key together with SHIFT the knob will backup
one page. You can reset the knob position when you are dragging the knob
with the mouse by clicking on the right mouse button.

*   Listview – Pressing the key will select the next entry in the list
scrolling the entries when necessary. Pressing the key together with
SHIFT will select the previous entry. Read only listview objects will
simply scroll the list forward and backward.

*   Cycle – Pressing the key will cycle one choise forward. Pressing the
key together with SHIFT will cycle one choise backward. Selecting the
gadget with the mouse while holding down the SHIFT key will also make the
gadget cycle backward. When the cycle gadget was setup to be a popup
gadget by the application programmer you can click inside the label area
to pop-up a menu from which you can select the desired item. You can
recognize a popup gadget by the imagery left of the label area. Normal
cycle gadgets have a circular arrow here and popup gadgets have a small
menu here.

*   Slider – Pressing the key will advance the knob position by one.
Pressing the key together with shift will decrease the knob position by
one. You can reset the knob position when you are dragging the knob with
the mouse by clicking to on the right mouse button.

*   String,Integer – Pressing the key will simply activate the gadget.

*   Mx – Pressing the key will select the next button skipping any
disabled buttons. Pressing the key together with shift will select the
previous button skipping any disabled buttons. If the Mx object is in
tabs mode the choises will not stop at the last or first selection but
instead it will cycle back to the start or the end.

   As you can see this behaviour is mostly style guide compliant with some
additional possibilities.

1.9   bgui.guide/Disclaimer

Disclaimer
****

   THERE IS NO WARRANTY FOR THE PROGRAMS, TO THE EXTENT PERMITTED BY
APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS

1.10   bgui.guide/License

```
                  License
*******
```

    This license applies to the product called BGUI, a program for the Amiga
computer, published by Jan van den Baard under the concepts of GiftWare, and
the accompanying documentation, example files and anything else that comes
with the original distribution. The term "Program" below, refer to this
product. The term "Author" below, refers to "Jan van den Baard". The license
is addresses as "you".

   * You may copy and distribute verbatim copies of the program's executable
     code and documentation as you receive it, in any medium, provided that
     you conspicuously and appropriately publish only the original, unmodified
     program, with all copyright notices and disclaimers of warranty intact
     and including all the accompanying documentation, example files and
     anything else that came with the original.

   * You may not copy and/or distribute this program without the accompanying
     documentation and other additional files that came with the original. You
     may not copy and/or distribute modified versions of this program.

   * You may not copy, modify, sublicense, distribute or transfer the program
     except as expressly provided under this license. Any attempt otherwise
     to copy, modify, sublicense, distribute or transfer the program is void,
     and will automatically terminate your rights to use the program under
     this license. However, parties who have received copies, or rights to
     use copies, from you under this license will not have their licenses
     terminated so long as such parties remain in full compliance.

   * By copying, distributing and/or using the program you indicate your
     acceptance of this license to do so, and all its terms and conditions.

   * Each time you redistribute the program, the recipient automatically
     receives a license from the original licensor to copy, distribute and/or
     use the program subject to these terms and conditions. You may not
     impose any further restrictions on the recipients' exercise of the rights

granted herein.

* You may not disassemble, decompile, re-source or otherwise reverse
  engineer the program.

* You are free to use the programs. If, after an evaluation period of up to
  30 days, you find the programs useful enough to continue using them you
  are morally, not lawfully, obligated to send the author a gift that, you
  think, matches his efforts. Shareware or commercial usage of the programs
  require registering. Please refer to the see
               Preface
                section for more
  information on registering.

* If you wish to incorporate parts of the programs into other programs,
  write to the author to ask for permission. If you want to release your
  programs as being Public Domain, FreeWare or GiftWare you are allowed to
  include a copy of the "bgui.library" without the accompaning
  documentation and files. ShareWare and Commercial distribution of the
  programs is not allowed without prior written permission. Please refer to
  the see
               Preface
                section for more information on registering.

* You agree to cease distributing the program and data involved if
  requested to do so by the author.

* You may charge a fee to recover distribution costs. The fee for diskette
  distribution may not be more than the cost to obtain a public domain
  diskette from Fred Fish.

1.11   bgui.guide/Future

Future
******

   This is a list of known bugs and quirks. Please read this before you start
sending me bug reports and suggestions.

     BUG = A bug that is still present.
     SUG = A suggestion which might be implemented in the future.
     REV = Revision and/or fine-tuning that still needs to be done.
     MAY = Maybe...

*   BUG - The layout engine locks up when, due to a bug or hacking, you
    re-size a window smaller than it's minimum as decided by BGUI. Reported
    by Florian Vorberger.

*   BUG - The infoclass screws up computing it's minimum dimensions when
    the last thing in the text is a command sequence. Should not be a real
    problem since it is bullshit to end a string with a command sequence.

*   BUG - There seem to be problems when using the textfield.gadget in a

BGUI context. The problem is that if you select an object while the
textfield object is active you get a double message from the clicked
object. I do not know why this is happening.

*   REV - BGUI gadgets on intuition requesters is NOT supported yet.

*   REV - The commodity class could use some cleaning up. I'm not really
satisfied with it as it is.

*   REV - The format of the library and class documentation files suck. I
should convert them to "real" autodoc format or amigaguide files.

*   REV - Error reporting. It's OK that a window failed to open or an
object failed to create but it would be nice to know why. Both numeric
and textual error descriptions should be made available.

*   REV - Adding method, conditional and hook notification to the window
class to. This will allow notification to happen on the application task
instead of the input.device task.

*   REV - Work on the menus. For now it is not possible to dynamically
add items, add notification or help files to the menus. This really sucks.

*   REV - Try to cut down on the code size. Although, all things
considering, BGUI is not excessive in size I'de like to see it smaller.

*   MAY - Letting the window backfill hook optionally make use of the
preferred workbench window backfill pattern or picture.

*   MAY - Fonts and images in the info class.

*   MAY - Add a posibility to "snapshot" window sizes to disk. Should
this be up to the application programmer? Suggested by Paul Weterings.

*   MAY - Take the layout engine of BGUI and make a seperate shared/link
library out of it. This would result in a +/- 10 KByte shared/link
library with the same functionality as the BGUI layout engine. Maybe this
will satisfy those people who are always bitching and complaining about
GUI engine libraries.

1.12   bgui.guide/Thanks To

Thanks To
****

   The author would like to extend a big thank you to the following people for
their valuable help:

   Bjorn Reese, Jim Cooper, Jim Karczewski, Keith Christopher, Michael
Wiedmer, Michiel Willems, Magne Oestlyngen, Magnus Enarsson, Martin Bischoff,
Parrish Heywood, Patrick van Beem, Paul Hill, Paul McLachlan, Paul Weterings,
Per Harald Myrvang, Ranulf Doswell, Zachary A Forsyth.

   And the people I may have forgotten...

Thanks should also go to:

* Gary Duncan, Heiko Rath and Boris Jakubaschk for giving me permission to
  use their SCSI-2 code for the BGUIPlayer demo program.

* All the folks at O.I.C. Obvious Implementations Corporation. This
  software is completely developed using DICE in the registered version and
  later on the commercial version.

* Francesco Devitt for MemLeak. A remarkable simple but very useful tool ;)

* Reinhard Spisser and Sebastiano Vigna for their wonderful Makeinfo which
  was used to generate this AmigaGuide®/ASCII documentation.

* Bryce Nesbitt and Michael Sinz for Enforcer. The best bug-hunter for the
  Amiga.

1.13 bgui.guide/Index

```
              Index
*****


              Disclaimer
                                Disclaimer

              Features
                                Features

              Future
                                Future

              Installation
                                Installation

              Introduction
                                Introduction

              Keyboard & Mouse
                            Keyboard & Mouse

              License
                                License

              Preface
                                Preface

              Read Me
                                Read Me

              Requirements
                                Requirements
```

Thanks To

Thanks To

Thanks To