Triton

An object oriented GUI creation system

Release 1.2

1 Introduction

1.1 Introduction

Triton is a standard Amiga shared runtime library. Triton makes it much easier to create good looking graphical user interfaces (GUIs) than GadTools, BOOPSI or other systems.

The most important features are:

- Object oriented system
- Automatically font sensitive, font adaptive
- Automatic keyboard shortcuts for default window actions
- *Really* easy to use
- Beautiful customizable AmigaOS3.x look
- Comes as a freely distributable shared library
- Size! Though Triton is very powerful, it is a shared library of less than 50KB. You don't need any startup tools, BOOPSI classes or other things. It's the Triton among the minnows of GUI creation systems ;)
- Resizability of windows wherever applicable
- A Preferences editor which allows you to customize the look and feel of all Triton GUIs

By using Triton you don't have to worry about otherwise very time-consuming things like font-sensitivity and resizability of your windows. What is even more important is that you can easily change your user interfaces later without having to rearrange display objects. Simply add an object to a group and the whole GUI will adapt to make room for it.

1.2 Copyright

The programs and files in this distribution are freely distributable, but are also © by Stefan Zeiger. They may be freely distributed as long as no more than a nominal fee is charged to cover time and copying costs. 'triton.library' is distributed as Freeware.

The Triton Preferences editor is shareware. If you want to customize the look and feel of Triton GUIs or if you simply like Triton and want to support its development, you have to register. You will then get a keyfile which will enable you to save all settings in the Preferences editor. In unregistered versions only the system options are saved.

If you want to register, please send DM20, US$15 or an equal amount in any other valid currency to me. See Section 3.3 [Odds & Ends - Support], page 9, for addresses.

Send money by EuroCheck (only in German Marks (DM)!), postal(!) money order or cash! It's impossible for me to cash in foreign checks, even if the amount is in DM. Please state the version of triton.library you have (use the 'C:Version' command to display the version).

There is absolutely **no** obligation for you to register Triton. If you don't need the registered Preferences editor, then you need not register.

Note: Registered trademarks are not explicitly marked as such in this manual. If a name is not marked as a trademark, this does not imply that it is free.

See Section 3.5 [Odds & Ends - License], page 11, for information about using Triton in your own - commercial or free - programs.

1.3 Installation

First of all, Triton requires at least OS2.04. If you're still running 1.2 or 1.3, you have to upgrade to OS2.04 or better. OS1.3 is *obsolete* and I will not support it. You certainly won't regret upgrading to OS2.1 or even better OS3.1.

The user relevant parts of Triton (i.e. the contents of the user archive) can be installed comfortably using the supplied installer scripts. Since 'Installer' ist © by Commodore and in order to save space, it is not included in the Triton archive. 'Installer' is included in the AmigaOS system software beginning with OS release 3.0 and can also be found as a freely distributable archive on many BBSs and ftp sites.

If you're running OS2.0x, use the 'Install/Workbench_2.0' installer script. If you're running OS2.1 or better, use the script for your preferred language from the 'Install/Workbench_2.1+' drawer. Note that not all installation scripts are fully localized. Unlocalized strings will simply be displayed in English.

Two versions of triton.library are included in this distribution. The one in libs37 requires OS2.04 or better, the one in libs39 requires OS3.0 or better and is optimized for OS3.0+, so that it is shorter and will run slightly faster than the OS2.04 version on an OS3.0+ system. The OS3.0+ version cannot be used with OS2.0/2.1.

1.4 Usage

The usage of a Triton GUI should be pretty clear. All windows have two additional keyboard shortcuts (if not explicitly disabled by the application):

- 'Esc' will simulate the 'close window' gadget.
- 'Del' will simulate the 'resize window' gadget.

Furthermore, various objects react on some standard keyboard shortcuts:

- In palette, slider, scroller and similar gadgets with an up/down facility you can use the shifted shortcut to decrease the value.
- Listviews can be controlled with either the cursor keys (`Shift`ed and `Alt`ernate shortcuts are also available) or the numeric keypad (including the numeric arrow keys, `Home`, `End`, `PgUp` and `PgDn`). The programmer may choose to disable these shortcuts, mainly in order to make the primary listview accessable via the cursor keys and the secondary listview via the numeric keypad.
- Within string gadgets, `Tab` and `Shift Tab` will activate the previous/next string gadget as usual. You can access all keyboard shortcuts by pressing them together with the right Amiga key (`RAmiga`). If no such shortcut exists, Triton will instead try to interpret the key you pressed as a menu shortcut. You may use the cursor up and down keys normally from within string gadgets to control a listview.

For information on how to use Triton as a programmer for developing Triton applications, see the developer documentation which is included in the Triton developer archive in the directory 'Developer'.

2 Preferences editor

2.1 Introduction

The Triton Preferences Editor allows you to customize the look and feel of applications which are using the Triton GUI creation system. The Preferences Editor is shareware. In unregistered versions only the system options are saved. See Section 1.2 [Introduction - Copyright], page 2, for details.

The Triton Preferences Editor allows you to change basically two kinds of settings:

1. Global settings for all Triton applications (e.g. public screens).
2. Private application settings.

Note that the listed global application `<<Global>>` belongs to the second type, too. If an application is created, it will inherit the global application's settings. Thus all applications without own settings will use the global settings. As soon as you change any setting of an application, it will become independant and will not inherit future changes of the global application. If you want to delete an application's private settings and make it use the global settings again (including inheriting future changes to them), select the menu item `Edit/Reset To Global`. This system may sound a bit complicated to you, but it is quite easy to handle.

2.2 Usage

Main window menu `Project`:

The Triton Preferences Editor can be installed by hand by simply dragging it into the 'Prefs' drawer of your boot partition ('`sys:Prefs`'). Then start the editor and confirm that you want to install the Preferences System. It may be removed again later by selecting 'Project/Remove' in the editor's main window's menu.

Main window menu `Edit`:

This menu contains all standard items plus `Reset to Global` (see Section 2.1 [Preferences editor - Introduction], page 5). All changes made from within this menu do only apply to the currently selected application.

Main window gadgets:

`Save`, `Use` and `Cancel` should be self-explaining. They apply to all modified settings of all applications. `Test` will restart the Preferences editor using the current settings without saving anything.

`Info` will give you all information available about the selected application. This includes name, version, compilation date and so on.

`Delete` will delete an application, i.e. its settings and information files. You have to restart the application to make it appear in the Triton Preferences Editor again. If you just want to delete an application's private settings, use `Edit/Reset to Global`.

The gadget `System` opens the system window which contains all global settings. All other gadgets open windows for private application settings.

`Frames` window:

The frames window allows you to configure the look of frames and groups. This should be mostly self-explaining.

`Images` window:

You can change some default Triton images (e.g. window backgrounds) in this window.

With `Use <Return> arrow in buttons` you can decide how buttons which can be activated by pressing `<Return>` will be highlighted. If you check the gadget, these buttons will have an arrow on the right side. By default they will have their label displayed in **bold** text instead.

`Pens` window:

In this window you can change some Triton pens. Especially important are `Halfshine` and `Halfshadow` which are used for rendering frames in `XEN 1`, `XEN 2` and `NeXT` look.

Also listed among these pens are the foregrounds and backgrounds of string gadgets. From a logical point of view the backgrounds should rather be images than pens but due to limitations of AmigaOS this is technically not possible.

`Windows` window:

This window contains the Triton window manager. The list shows all windows of the selected application which have been opened since the last reboot or are saved permanently to 'envarc:Triton'. The windows are displayed by their titles (if available; otherwise the ID number in square brackets). If you close a Triton window, Triton will remember its last dimensions

in a file in 'env:Triton', so that it will have the same dimensions when you reopen it. If you want to keep the window dimensions permanently, so that they are always available (unless you change them later on), select the appropriate window and click on `Snapshot`. This system works just like the window snapshotting of the Amiga Workbench. If you want Triton to "forget" a custom window position and reposition the window according to the original positioning information (e.g. centering the window on the screen after choosing a new screen resolution), you can `UnSnapshot` the window. The gadgets `Snapshot all` and `UnSnapshot all` will (un)snapshot all windows of the currently selected application. The gadget `Info` will bring up a requester containing some information about the currently selected window.

`System` window:

The system window contains all global settings. The most important one of those is the public screen manager. You can add, delete and edit public screens. Double click on a screen or select `Use` and the name of the screen will be copied into the above string gadget. This is the screen for the currently selected application. You may also type in any other screen name (either of an existing public screen or of a public screen which is managed by the Triton public screen manager).

When Triton opens a window, it will first look if the specified screen exists and use it in this case. Otherwise it will try to create it if an entry for this screen can be found in the screen manager. If this doesn't work either, Triton will fall back onto the default public screen. You can test how a screen will look by selecting `Test`. The screen will be opened and can be closed again by closing the information window which is displayed in it. Note that the screen will not be made public when testing.

The screen manager settings override any specifications made by the programmer. Setting an application's screen to `<<Program default>>` will make it use the screen specified by the programmer. However only public and workbench screen windows can be promoted using the screen manager. Custom screen windows will always open on the custom screen specified in the application.

Except for the public screen manager, the `System` window currently contains only one switch: `Simple refresh`. Select it to make all windows use `Simple` instead of `Smart` refresh. This will require less memory but make the refresh slower und worse looking (at least with the standard AGA graphics - I haven't tried it with a graphics board).

3 Odds & Ends

3.1 Astronomy lesson

Triton - A moon of Neptune
(by Michael Berg)

Triton is a quaint little moon, in that it is one of the very few moons to be known to have retrograde orbits around their host. Triton is believed to be a 3-6000 km large world of mostly liquid nitrogen oceans, perhaps with a thin methane atmosphere as well, but noone knows for sure since either of the Voyager twins ever got close enough to take convincing snapshots.

Neptune itself is also quite strange in that it has an almost 90 degree tilted inclination, so that it "rolls" through space, unlike Earth and the other planets (minus Venus), which all spin like tops in their orbits. Noone can tell for sure why Neptune has such an eccentric inclination, but it has been suggested that it was caused by a gigantic collision of some sort (probably with a large asteroid). In the process Neptune also lost its third moon, which at the time was none other than Pluto. The collision accelerated Pluto to the point where it had enough momentum to actually leave its orbit and shoot into space on its own. As it did so it probably passed below the Roche' radius of Neptune, causing it to break up in two or more bodies under the influence of the massive tidal forces of Neptune. This would certainly account for Charon, which is in itself too large to have been "captured" by an object as small as Pluto.

Okay, end of today's astronomy lesson :-)

3.2 Acknowledgments

Thanks must go to (in random order):

- Nico François for his suggestions, beta-testing and helping me with some very weird problems in Triton.
- Carsten Raufuß for beta-testing, suggestions and nagging. ('When will you finally implement listviews? I do really need them for my GUI!' ;)
- Kai Iske for his BOOPSI class sources.
- Michael Berg for the 'Astronomy Lesson' and beta-testing.
- Magnus Holmgren for the Swedish translation and DICE support.
- Frank Verheyen for the Dutch translation and the AmigaE support.
- Gunther Nikl for the gcc support.
- Klaus Melchior for the DICE support.

- Peter Fröhlich for the AmigaOberon support.

- Oskar Liljeblad for the assembler support.

- Stefan Schulz for the M2Amiga support.

- All the guys on `SPOT.BETA` for beta-testing and suggestions.

- Philipp Lonke, Marco Frischkorn and Patrick Ohly for beta-testing.

- Danny Schrod, Chris Reichert and Bernd Hassenzahl (SysOps of `Plus`, `Publishers Treff` and `Tec Mania Systems`) for support.

- Tom Eicher for proof-reading the documentation.

- Commodore for the Amiga.

I have used the following software products for creating Triton:

- The C sources were compiled with *SAS/C 6.51*. With SAS/C it was possible to create Triton without a single line of assembly language.

- The sources and the documentation were edited with *GNU Emacs 18.59*.

- The documentation was formatted using *PasTeX* and *makeinfo*.

- The icon was painted with *IconEdit* and *Personal Paint*.

- Shell commands were entered in a *KingCON* window ;-)

3.3 Support

If you have suggestions or remarks about Triton, or if you find any bugs, please let me know.

Contacting the author:

- **EMail:**
  - s.zeiger@tecmania.tms.mc.org
  - stefan@sweet.rhein-main.de
  - Stefan.Zeiger@p15.f6302.n244.z2.fidonet.org

- **ADSP:**
  stefan@wwsp.adsp.sub.org

- **FidoNet:**
  Stefan Zeiger 2:244/6302.15

- **Mail:**
  Stefan Zeiger
  Seligenstädter Weg 24
  D-63796 Kahl
  Germany

- **Voice:**
  +49-6188-2525 (after 6:00 PM UT only)

Please use EMail/FidoNet if possible. This is more convenient for me and your chance of getting a reply quickly is much better.

If you had any problems reaching me at my old EMail addresses, please try again now with my `TecMania` address.

```
         __
      __///
      \\\/...if(2B|!2B) user->prefs.shakespeare=TRUE;
```

3.4 Updates

Beginning with release 1.2, the Triton archives are named '`triXXusr.lha`' (user archive) and '`triXXdev.lha`' (developer archive) with `XX` being the release number, e.g. '`tri12usr.lha`' and '`tr12dev.lha`' for Triton release 1.2. The complete archive, including both user and developer archives is called '`tritonXX.lha`'.

The latest release of Triton is always available via FidoNet filerequest from `Publishers Treff` BBS. The user archive of the latest Triton release can be requested as '`TRITON`'.

```
       Name : Publishers Treff
      SysOp : Christoph Reichert
    FidoNet : 2:244/6302.0
  Protocols : V32b (14k4), ZyX (19k2), V42b, 8N1
```

You may also log in as a guest at `Plus` BBS and at the BBS of `Tec Mania Systems` in order to download Triton directly. At `Plus`, Triton can be found in '`/WIZARDWORKS/PROGRAMME`', and in '`/T-M-S/BINAER/AMIGA/WIZARD-WORKS`' at `Tec Mania Systems`.

```
       Name : Plus
      SysOp : Daniel Schrod
      Phone : +49-6023-32603
  Protocols : V32b (14k4), ZyX (19k2), V42b, 8N1


       Name : Tec Mania Systems
      SysOp : Bernd Hassenzahl
      Phone : +49-6157-8742-7 (V34 28k8)
              +49-6157-8742-8 (V32t 19k2)
              +49-6157-91012-1 (ISDN X75)
              +49-6157-91012-2 (ISDN X75)
```

Updates of the Triton packages will furthermore be distributed in the following Amiga software networks:

- AmiNet (directory '`/dev/gui`')
- FrAS (directories '`/FRAS/BIN/TOOLS`' and '`/FRAS/BIN/PROGRAMMIEREN`')
- ADS (directory '`ADSWORK2`')

3.5 License

1. DISCLAIMER:

   THERE IS NO WARRANTY FOR THE TRITON USER AND DEVELOPER PACKAGES, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDER OR ANY OTHER PARTY WHICH MAY DISTRIBUTE THE PACKAGES PROVIDES THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

   IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL THE COPYRIGHT HOLDER OR ANY OTHER PARTY BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF THE COPYRIGHT HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

2. **Commercial and FD usage:**

   You may use Triton freely for all free applications (Freeware, Public Domain, Copyleft,...). If you want to use Triton for a shareware product, please register Triton and send me an uncrippled version of your product. If you want to use Triton for a commercial product, please contact me directly (see Section 3.3 [Odds & Ends - Support], page 9) for information about licensing Triton.

3. **Distribution:**

   All of the files copyrighted by the author must remain unmodified. None of these files may be distributed on its own, the entire package must be distributed as a whole. If not explicitly stated otherwise all files are subject to my copyright on the Triton package. All icons are excluded from this copyright. The Triton Preferences Editor icon has been created by myself but is hereby placed into the public domain with no restrictions whatsoever.

   There is one exception to the above redistribution limitations. If you plan to release a freely distributable program, you may include the Distribution drawer (which can be found under

the name 'Triton' in the '`Developer/Distribution`' directory) with your distribution. It is left to the programmer to decide whether he wishes to register Triton.

Whether your program is freely distributable or commercial, you must state in your documentation that your program uses triton.library and that Triton is © by Stefan Zeiger. A notice in the "About" requester saying "This application uses the Triton GUI creation system, which is © by Stefan Zeiger" wouldn't be bad, either ;)

Mr. Fred Fish and Mr. Martin Schulze are hereby explicitly granted the permission to distribute the Triton packages as part of their Fish and SaarAG disks and CDROMs.

Glossary

'`BOOPSI`'   Basic Object Oriented Programming System for Intuition. See the BOOPSI chapter of the RKM ("ROM Kernel Manual") volume "Libraries" for details.

'`DICE`'   Dillon's Integrated C Environment. A C development system for AmigaOS.

'`Endless loop`'
          See '`Loop, endless`'.

'`FD software`'
          Freely distributable software, including '`Public Domain (PD)`', '`Freeware`', '`Giftware`', '`Shareware`', etc.

'`GCC`'   '`GNU C`': The C compiler of the "Free Software Foundation". This compiler and its development system is freely distributable and a port is also available for AmigaOS.

'`GUI`'   Graphical User Interface.

'`Loop, endless`'
          See '`Endless loop`'.

'`SAS/C`'   A commercial C development system for AmigaOS.

Index

Table of Contents