

main

COLLABORATORS

	TITLE : main		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	main	1
1.1	Tricky Color System, tcs.library 1.69	1
1.2	I Disclaimer and Distribution	1
1.3	II Current Release Information	3
1.4	III Author	4
1.5	IV History	4
1.6	V Future	8
1.7	VI Greetings and Thanks	9

Chapter 1

main

1.1 Tricky Color System, tcs.library 1.69

Tricky Color System, tcs.library 1.69 (26.9.2000)
© 2000 Simone Bevilacqua

I	Disclaimer and Distribution	usual legal stuff
II	Current Release Information	IMPORTANT! read carefully!
III	Author	the madman behind
IV	History	what has happened
V	Future	what will happen
VI	Greetings and Thanks	ciao!
1	Introduction	description of TCS and its features
2	TCS Basics	familiarizing with TCS
3	How TCS Works	for techie-minded people
4	The tcs.library	writing programs that use TCS

1.2 I Disclaimer and Distribution

I Disclaimer and Distribution

```
*****
*THE SOFTWARE BELONGING TO THIS DISTRIBUTION IS PROVIDED "AS-IS" WITHOUT *
*WARRANTY OF ANY KIND EITHER EXPRESSED OR IMPLIED.                      *
*I ACCEPT NO RESPONSABILITY OR LIABILITY FOR ANY DAMAGE OR PROBLEM DE- *
*RIVING FROM THE USE OF ANY OR ALL OF ITS PARTS: USE AT YOUR OWN RISK    *
*****
```

This package is Freeware as long as it is used for Public Domain, Freeware or private-use-only software; for commercial use a written agreement with me is *necessary*.

Distribution on magazines' coverdisks and Aminet CDs is allowed.
Distributions must be free of charge (excluding the media/shipping/etc.
expenses) and must:

- leave every file unmodified
- include all the following files preserving the directory structure:

```

TCS
|
+---doc
|   |
|   +---basix.guide           documentation
|   +---intro.guide          documentation
|   +---lib.guide             documentation
|   +---main.guide            documentation
|   +---main.guide.info       icon
|   +---tech.guide            documentation
|
+---demos
|   |
|   +---readme                short information
|   |
|   +---exe
|       |
|       +---DubBuf            example executable
|       +---DubBuf&FRScrl     example executable
|       +---DubBuf&HRScrl     example executable
|       +---FullRes           example executable
|       +---HalfRes           example executable
|       +---ILBM&Fade         example executable
|       +---RmpILBM           example executable
|       +---SvRGBxPal         example executable
|       +---TriBuf            example executable
|       +---TriBuf&BltFRP     example executable
|       +---TriBuf&Zm         example executable
|       +---XPfld             example executable
|
+---inc
|   |
|   +---dat.i                 example include source
|   +---macros.i             example include source
|   +---shl_strtup.i          example include source
|
+---pix
|   |
|   +---Alphonse.rgbh.iff     example picture
|   +---brshs.rgbh.iff        example picture
|   +---DXP.rgbh_15.iff       example picture
|   +---msk.rgbs.iff          example picture
|   +---MyWB.rgbs.iff         example picture
|   +---Suppa.rgbs.iff        example picture
|   +---TigerMask.iff         example picture
|   +---TigerMask.rgb332.iff  example picture
|   +---TigerMask.rgbs.iff    example picture
|   +---TigerMask.rgbw.iff    example picture
|   +---wheel.rgbw.iff        example picture

```

```

|   |
|   \---cod
|       |
|       +---do                simple script for compiling
|       +---DubBuf.s          example source
|       +---DubBuf&FRScrl.s   example source
|       +---DubBuf&HRScrl.s   example source
|       +---FullRes.s         example source
|       +---HalfRes.s         example source
|       +---ILBM&Fade.s       example source
|       +---RmpILBM.s         example source
|       +---SvRGBxPal.s       example source
|       +---TriBuf&BltFRP.s   example source
|       +---TriBuf&RotZm.s    example source
|       +---TriBuf&Zm.s       example source
|       +---TriBuf.s          example source
|       \---XPfld.s          example source
|
+---includes
|   |
|   \---libraries
|       |
|       +---tcs.i             assembly include
|       \---tcs_lib.i        assembly include
|
+---libs
|   |
|   \---tcs.library          library binary
|
\---pal
    |
    +---readme               short information
    +---RGB332.iff           palette
    +---RGBH.iff             palette
    +---RGBM.iff             palette
    +---RGBP.iff             palette
    +---RGBS.iff             palette
    \---RGBW.iff             palette

```

1.3 II Current Release Information

II Current Release Information

The previous releases included `_early_/_preliminary_` versions of the library and support files, which are **not** binary nor source compatible with the present one.

Therefore, `_users_` should **NOT** use the **new** version of the library with applications using the **old** ones.

`_Developers_`, to easily update their programs, should always use properly the enclosed include files and should never make any "special" assumption: in this way, applications could need just to be recompiled; to proceed

correctly, however, it is recommendable to read carefully the list of changes in the history.

This release is still in `_preliminary_` stage, so keep all these advices in mind.

1.4 III Author

III Author

Hi there!

I `*do*` want your feedback.

Let me know what you think and if you have any problems/ideas or need some explanations/hints.

You can write to:

bevilacq@cli.di.unipi.it

(this address will be valid only until the end of September)

if you want to use the old (and not that trusty) snail-mail:

Simone Bevilacqua
Via A.Volta 6
86010 Ferrazzano (CB)
ITALY

(the answer could take a bit... please be patient!)

1.5 IV History

IV History

1.69 (26.9.2000)

While preparing a demo proggy for the exam, I had another total brain-storm, so the library underwent some serious revision - not to mention the additions...

[still a PRELIMINARY release: do `*not*` expect it to be compatible with other versions]

- * due some of the following changes many LVOs have changed!

- * functions "grouped" differently

- * functions renamed:

 - * `TCS_WrtTxtr()` -> `TCS_WrtBrsh0()`

```
* TCS_FitTxtr1()    -> TCS_FitBrshZn0()
* TCS_FitTxtr4()    -> TCS_FastFitBrshZn0()
* TCS_RotZmTxtr1()  -> TCS_RotZmBrsh0()
* TCS_RotZmTxtr4()  -> TCS_FastRotZmBrsh0()
* TCS_CpyScr        -> TCS_CpyScr0()
("texture(s)" renamed as "brush(es)" everywhere and thus also the string
"Txtr" appearing in function/argument names has been replaced by "Brsh")
* functions added:
  - TCS_GetBrsh()
  - TCS_MkBrshMsk()
  - TCS_MrgBrshs()
  - TCS_MixBrshs()
  - TCS_FreeBrsh()
  - TCS_WrtBrsh()
  - TCS_WrtBrsh1()
  - TCS_WrtBrsh8()
  - TCS_WrtBrsh9()
  - TCS_FastWrtBrsh()
  - TCS_FastWrtBrsh0()
  - TCS_FastWrtBrsh8()
  - TCS_WrtBrshZn()
  - TCS_WrtBrshZn0()
  - TCS_WrtBrshZn1()
  - TCS_WrtBrshZn8()
  - TCS_WrtBrshZn9()
  - TCS_FastWrtBrshZn()
  - TCS_FastWrtBrshZn0()
  - TCS_FastWrtBrshZn8()
  - TCS_FitBrsh()
  - TCS_FitBrsh0()
  - TCS_FitBrsh8()
  - TCS_FastFitBrsh()
  - TCS_FastFitBrsh0()
  - TCS_FastFitBrsh8()
  - TCS_FitBrshZn()
  - TCS_FitBrshZn8()
  - TCS_FastFitBrshZn()
  - TCS_FastFitBrshZn8()
  - TCS_RotZmBrsh()
  - TCS_RotZmBrsh8()
  - TCS_FastRotZmBrsh()
  - TCS_FastRotZmBrsh8()
  - TCS_CpyScr()
  - TCS_CpyScr1()
  - TCS_FastCpyScrZn()
* functions removed:
  * TCS_FitTxtr2()
  * TCS_RotZmTxtr2()
* functions modified:
  * TCS_WrtBrsh0(): input arguments changed; optimized
  * TCS_FlgDXPfldCols(): flagging inverted
  - TCS_SetFPfldOpct(): Dual Cross Playfield part optimized a bit
  - TCS_ShwdSpl() and TCS_HideDspl(): for automatic hiding of previously
    shown displays
  - LdILBM(): now accepts the mskHasTransparentColor compression flag set
    (all it does is ignoring it)
- functions fixed:
```

- TCS_CpyScrZn?(): bad positioning after source top-left clipping
- TCS_EnbXPfld(): movem on exit pulled a long extra from stack
- TCS_EquVdoBufs(): didn't copy VdoPln1 in FullRes only
- TCS_EnbDXPfld(): didn't update displays status correctly when called on back playfield
- TCS_SetFPfldOpct(): performed its operations on a register which was never initialized before (Dual Cross Playfield only) - it's a mystery how sometimes it managed to work 100% fine!
- Graphic Context masking mode added (TCS_GCb_msk, TCS_GCf_msk))
- * tcsb_ShwnDIAdr added to library base structure
- examples:
 - revised & adapted to new library
 - brush handling example (brshs.s) added
 - TriBuf&RotZm.s extended for masked rotzooming
 - XPfld.s fixed (didn't set dl.b for SetFPfldOpct() - this is because I forgot to add this parameter in the guide!)
 - pictures renamed
 - "MyWB.rgbm.iff" (formerly "WB.rgbm.iff") picture updated
 - brushes' picture ("brshs.rgbh.iff") added
 - minor changes to support files
- usual additions/changes/corrections to documentation (especially to the graphic functions and RGBx technical sections)

1.68 (8.9.2000)

Finally I managed to force myself to work (but many weeks have been wasted anyway) and pull out this major update.

[still a PRELIMINARY release: do *not* expect it to be compatible with other versions]

[for some stupid reasons I failed to upload this version to Aminet, so it has never been publically released]

* due some of the following changes many LVOs have changed!

* functions renamed:

- * TCS_GetRGBxBrtns() -> TCS_GetRGBxTheoBrtns()
- * TCS_CnvRGB() -> TCS_RGBToRGBx()
- * TCS_CnvRGBx() -> TCS_RGBxToRGB()
- * TCS_GetVdoBufs() -> TCS_GetVdoBufsAdrs()

* functions added:

- TCS_C2PPass0()
- TCS_EquVdoBufs()
- TCS_GetRGBxActlBrtns()
- TCS_PltPx14()
- TCS_PltPx15()
- TCS_DrwLn4()
- TCS_DrwLn5()
- TCS_DrwHrzLn4()
- TCS_DrwHrzLn5()
- TCS_DrwVrtLn4()
- TCS_DrwVrtLn5()
- TCS_DrwFrm4()
- TCS_DrwFrm5()
- TCS_DrwFrm6()
- TCS_DrwFrm7()
- TCS_DrwSqr4()
- TCS_DrwSqr5()

- TCS_DrwSqr6()
- TCS_DrwSqr7()
- TCS_DrwTrngl4()
- TCS_DrwTrngl5()
- TCS_DrwTrngl6()
- TCS_DrwTrngl7()
- TCS_DrwPlgn4()
- TCS_DrwPlgn5()
- TCS_DrwPlgn6()
- TCS_DrwPlgn7()
- TCS_DrwOpnPlgn4()
- TCS_DrwOpnPlgn5()
- TCS_DrwCrcl4()
- TCS_DrwCrcl5()
- TCS_DrwCrcl6()
- TCS_DrwCrcl7()
- TCS_FillArea4()
- TCS_FillArea5()
- TCS_CpyScrZn()
- TCS_CpyScrZn0()
- TCS_CpyScrZn1()
- TCS_WrtTxtr()
- TCS_RotZmTxtr1()
- TCS_RotZmTxtr2()
- TCS_ClpLn()
- * functions modified:
 - * TCS_GetRGBxTheoBrtns(): RGBH support (I forgot this!) added; return value size changed (.b -> .w)
 - * TCS_RGBToRGBx(): RGB->RGBW conversion totally rewritten (was wrong!); finished (all RGBx modes are supported now)
 - TCS_DrwCrcl2(): optimized
 - TCS_RotZmTxtr4(): finished
 - TCS_LdILBM(): uncompressed ILBMs support added
- functions fixed:
 - TCS_DrwSqr?(): trashed d2 & d3
 - TCS_FillArea?(): trashed a3 & could GURU
 - TCS_GetRGBxTheoBrtns(): various
 - TCS_LdILBM(): small
 - several Graphic Context -sensitive functions were not using the appropriate routines (I forgot to update their jump-tables)
- * TCS_SetRGBxMode() moved among the display functions
- * TCS_FillBuf() moved among the special graphic functions
- filled polygons functions finished
- all color functions finished
- * TCS_GBb_eor/TCS_GBf_eor renamed as TCS_GBb_inv/TCS_GBf_inv
- Graphic Context inverse drawing support added
- * constants called TCS_D_#? renamed as TCS_C_#?
- * maximum RGBx index (TCS_VM_RGBx) changed from 255 to 127
- examples:
 - XPfld.s fixed (didn't set the right palette for the front playfield after activating the Dual mode)
 - RmpILBM.s improved & fixed
 - DubBuf&HRScr1.s, DubBuf&FRScr1.s and TriBuf&RotZm.s added
 - all sources slightly revised
 - some pictures removed/substituted/added
- usual additions/changes/corrections to documentation (especially to the basics and RGBx technical sections)

1.67 (26.6.2000)

Very little work in the last two months, so this version isn't a great improvement over the previous one.

[still a PRELIMINARY release: do **not** expect it to be compatible with other versions]

- * changes in tcs.i:
 - * TCS_II_BufAdr renamed as TCS_II_UsrBufAdr
- * changes in tcs_lib.i:
 - * several LVOs changed
 - * TCS_SetRGBx() renamed as TCS_SetRGBxMode()
- Blitter-assisted FullRes conversion optimization (thanks to Antonello)
- additions/changes/corrections to documentation

1.66 (13.4.2000)

First public release.

[PRELIMINARY: compatibility **not** guaranteed with future versions]

- entries marked with an '***' are relative to compatibility issues

1.6 V Future

V Future

Things I've got in mind for the next versions:

- implementing the missing functions
- adding TCS_BltFRPass1(), TCS_C2PPass1()
- adding a complete set of functions for fonts handling
- improving/completing/correcting/extending this documentation (more information about structures, more programming hints, etc.)
- writing autodoc & FD
- fixing the known bugs
- (?) making MskPln compulsory
- (?) extending TCS_SetPlnsPos() for automatic and SHRES scrolling of FullRes screens
- (?) making the library more OS-friendly
- (?) adding new video modes
- (?) almost complete re-writing of the graphic functions

the question is: will I ever have enough time and, above all, how much longer will I keep on developing this project (probably I'll stop soon - in this case I'll either give the sources to anybody willing to continue it, or I'll publically distribute them)??

- some people asked if a standard video driver would be possible... the answer is both yes and no: with a totally OS-friendly version of the library (which *can* be done) writing a video driver should be possible and even rather easy (right now I ignore how to write a video driver, so this is just a supposition); the problem would be that all the functions of the graphics.library are not written for use on chunky screens (one of the problems of graphic cards), so the video driver would be useless
- another thing I've been asked about is 31 kHz screens: since TCS screens require 35 ns (SHRES) pixels, that refresh rate cannot be achieved due to hardware limitations
- possible new video modes: 16-bit, 24-bit and 105 ns pixels ("MaxRes"); all of these video modes are quite CPU expensive, so, before starting to work on them, I've still got to think more about them. 16-bit is quite feasible and maybe one day it will see the light; 24-bit would be probably worthless, as its color definition grain would be wasted by the nature of TCS itself; MaxRes video mode (which appeals me a lot!) needs a little introduction: the idea is to use just 3 SHRES pixels (instead of 4 - read here for more information) for a TCS pixel, in order to increase the horizontal resolution; yet, pixels would look darkish (the three components would give at most 1/3 of the maximum brightness possible) and the maximum number of colors would be just 64, (just 3 bits per bitplane); besides, the chunky2planar conversion would be more difficult (and thus probably slower) to perform, as groups of 3 bits aren't easy to treat with bytes, bit shifts and even accesses to CHIP memory; on the other hand, it would not require any adaption of the internal palette routines (it would be totally compatible with all the RGBx modes) and graphic functions (the chunky pixels would still be 8-bit - though two of them would be unused)

1.7 VI Greetings and Thanks

VI Greetings and Thanks

I won't bother you with an oversized list, but there are some people I really owe some particular thanks:

- Victor Haaz for his invaluable support, suggestions and nice e-mailing
- my professor Giuseppe Attardi for accepting TCS as project for the exam of his Computer Graphics course
- all the people who sent me their feedback responses

Greetings fly to everyone who knows me.
