DISPLAYLIB

Conversion program

COLLABORATORS

	TITLE : DISPLAYLIB		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

DISPLAYLIB

1.1 Overview of DISPLAYLIB

```
                          Overview


                   An Acid Software Library

                  Converted to AmigaGuide by

                    Red When Excited Ltd

             Used with the permission of Acid Software

        Edited, fixed and cleaned by Toby Zuijdveld 26/02/1999.
                    mailto:hotcakes@abacus.net.au
```

1.2 DISPLAYLIB

```
Statement: CopList
--------------------------------------------------------------------------------
Modes  :
Syntax : CopList
```

1.3 DISPLAYLIB

```
Statement: InitCopList
--------------------------------------------------------------------------------
Modes  :
Syntax : InitCopList CopList#[,type]|[,ypos,height,type,sprites,colors,numcustoms]

InitCopList is used to create a CopList for use with the CreateDisplay
```

command. The ypos, height parameters define the section of screen.
Sprites, colors and customs will allocate instructions for that many
sprites (always=8!) colors (yes, as many as 256!) and custom copper
instructions (to be used by the new DisplayFX library currently in
devlopment).

The widthadjust parameter is currently not implemented, for display
widths other than standard see the DisplayAdjust command. The following
constants make up the type parameter, add the number of bitplanes to
the total to make up the type parameter.

```
Flag          Value
#smoothscroll $10
#dualplayfield  $20
#halfbrite  $40
#ham          $80

#lores         $000
#hires         $100
#super         $200

#loressprites $400
#hiressprites $800
#supersprites $c00

#fmode0        $0000
#fmode1        $1000
#fmode2        $2000
#fmode3        $3000

#agapal        $10000
```

For displays on non-AGA machines only #fmode0 and #loressprites are
allowed. More documentation, examples and fixes will be published soon
for creating displays.

1.4 DISPLAYLIB

Statement: CreateDisplay

Modes :
Syntax : CreateDisplay CopList#[,CopList#..]

CreateDisplay is used to setup a new screen display with the new
display library. Any number of CopLists can be passed to CreateDisplay
although at present they must be in order of vertical position and not
overlap. CreateDisplay then links the CopLists together using internal
pointers, bitmaps, colours and sprites attached to
coplists are not affected.

1.5 DISPLAYLIB

```
Statement: DisplayBitMap
--------------------------------------------------------------------------------
Modes  :
Syntax : DisplayBitMap CopList#,bmap[,x,y] [,bmap[,x,y]]
```

The DisplayBitMap command is similar in usage to the slice libraries'
show commands. Instead of different commands for front and back
playfields and smooth scroll options there is only the one
DisplayBitMap command with various parameter options. With AGA
machines, the x positioning of lores and hires coplists uses the
fractional part of the x parameter for super smooth scrolling.The
CopList must be initialised with the smooth scrolling flag set if the
x,y parameters are used, same goes for dualplayfield.

1.6 DISPLAYLIB

```
Statement: DisplaySprite
--------------------------------------------------------------------------------
Modes  :
Syntax : DisplaySprite CopList#,Sprite#,X,Y,Sprite Channel
```

DisplaySprite is similar to the slice libraries ShowSprite command with
the added advantage of super hires positioning and extra wide sprite
handling. See also SpriteMode and the Usage discussion above.

1.7 DISPLAYLIB

```
Statement: DisplayPalette
--------------------------------------------------------------------------------
Modes  :
Syntax : DisplayPalette CopList#,Palette# [,coloroffset] [,coloroffset,bankoffset]
```

DisplayPalette copies colour information from a Palette to the CopList
specified.

1.8 DISPLAYLIB

```
Statement: DisplayControls
--------------------------------------------------------------------------------
Modes  :
Syntax : DisplayControls CopList#,BPLCON2,BPLCON3,BPLCON4
```

DisplayControls allows access to the more remote options available in
the Amiga's display system. The following are the most important bits

from these registers (still unpublished by Commodore!*()@GYU&^)

The default values are given at the top of the table, the parameters
are exclusive or'd with these with DisplayControls so to set all the
sprite color offsets to 1 so that sprite colours are fetched from color
registers 240..255 instead of 16..31 we use...

DisplayControls 0,0,0,$ee

bit BPLCON2=$224 BPLCON3=$c00   BPLCON4=$11

```
15  *          BANK2 * active colour bank  BPLAM7 xor with bitplane
14  ZDBPSEL2  BANK1 *                      BPLAM6 DMA for altering
13  ZDBPSEL1  BANK0 *                      BPLAM5 effective colour
12  ZDBPSEL0  PF2OF2 coloffset pfield 2 BPLAM4 look up
11  ZDBPEN        PF2OF1                   BPLAM3
10  ZDCTEN        PF2OF0                   BPLAM2
09  KILLEHB * LOCT *palette hi/lo nibble  BPLAM1
08  RDRAM=0 *                             BPLAM0
07  SOGEN         SPRES1 *sprites resolution  ESPRM7 high order color
06  PF2PRI H  SPRES0 *                 ESPRM6 offset for even
05  PF2P2         BRDRBLANK border is black ESPRM5 sprites
04  PF2P1         BRDNTRAN border hits ZD      ESPRM4
03  PF1P0                                 OSPRM7 high order color
02  PF1P2         ZDCLCKEN ZD=14Mhz clock      OSPRM6 offset for odd
01  PF1P1         BRDSPRT  sprites in borders!  OSPRM5 sprites
00  PF1P0         EXTBLKEN wo blank output? OSPRM4
```

```
 ! – Don't touch
 H – See standard hardware reference manual
 * – controlled by display library
ZD – any reference to ZD is only a guess (just sold my genlock)
```

1.9 DISPLAYLIB

Statement: DisplayAdjust

Modes :
Syntax : DisplayAdjust CopList#,fetchwid,ddfstrt,ddfstop,diwstrt,diwstop

Temporary control of display registers until I get the widthadjust
parameter working with InitCopList. Currently only standard width
displays are available but you can modify the width manually (just
stick a screwdriver in the back of your 1084) or with some knowledge of
Commodores AGA circuitry.

Ha ha ha. No to be quite serious I really do not have a clue how they
cludeged up the Amiga chip set. When ECS was introduced suddenly all
display fetching moved to the right. Now they seem to have done the
same to sprites so it is near impossible to have them all going without
limiting yourself to a seriously thin display.

If you hack around with the system copperlists you'll find they

actually change fetch modes as you scroll a viewport across the display
and commodore say you should not use sprites anyway so as to be
compatible with their new hardware which is rumoured to run WindowsNT,
yipeee. By then we will be hopefully shipping the jaguarlib for
Blitz2.

Advanced Usage

The following is the coplist object newtype. Advanced users may wish to
access some of the pointers so as to directly modify the copper
list.

User workspace within a copper list is available via the customs
parameter in the InitCopList command. The customs pointer within the
object will then point to the area in memory which is initialised with
$1fe0000 which is the equivalent of a NOP instruction for the
copper.

```
NEWTYPE .mycop
 size.l       ;0 = not initialised
 coppos.l     ;location in chipmem
 colors.l
 sprites.l
 bpcons.l
 bplanes.l
 dot.l
 customs.l
 dob.l
 numbp.w:colpokes.w          ;limits
 fetchwid.w:xand:xshift      ;for show calculations (3 words)
 ypos.w:height:res
 numsprites.w:numcols:numcustoms
 aga.w                 ;24bit=$8000 fetch = $00,$10,$20,$30
 resshift.w            ;lo,hi,shi = 2 1 0
 setup.w               ;lines taken for setup
 cblow.w               ;if custom goes below 256
 sfetch.w:spres:spif:spwid:sspwid  ;sprite mode for display
End NEWTYPE
```


1.10 DISPLAYLIB

Statement: CustomColors

Modes   :
Syntax : CustomColors CopList#,CCOffset,YPos,Palette,startcol,numcols[,thru256flag ↩
    ] ;ecs=2+n aga=2+n+n+n/16

Using the custom copper space in a display, CustomColors will alter the
displays palette at the given YPos. The number of customcops required
is either 2+numcols for ecs displays and 2+n+n+n/16 for aga displays.
In aga, numcols must be a multiple of 32.

Note that large AGA palette changes may take several lines of the
display to be complete.

1.11 DISPLAYLIB

```
Statement: CustomString
-------------------------------------------------------------------------
Modes  :
Syntax : CustomString CopList#,CCOffset,YPos,Copper$  ;2+n {n=len(a$)/4}
```

1.12 DISPLAYLIB

```
Statement: DisplayDblScan
-------------------------------------------------------------------------
Modes  :
Syntax : DisplayDblScan CopList#,Mode[,copoffset]   ;2
```

DisplayDblScan is used to divide the vertical resolution of the display
by 2,4,8 or 16 using Modes 1,2,3 and 4. This is most useful for fast
bitmap based zooms. A Mode of 0 will return the display to 100%
magnification.

As with the DisplayRainbow, DisplayRGB, DisplayUser and DisplayScroll
commands DisplayDblScan uses the new line by line copper control of the
display library. To initialise this mode a negative parameter is used
in the CustomCops parameter of the InitCopList command. DisplayDblScan
requires 2 copper instructions per line (make CustomCops=-2).

1.13 DISPLAYLIB

```
Statement: DisplayRainbow
-------------------------------------------------------------------------
Modes  :
Syntax : DisplayRainbow CopList#,Register,Palette[,copoffset] ;ecs=1 aga=4
```

1.14 DISPLAYLIB

```
Statement: DisplayRGB
-------------------------------------------------------------------------
Modes  :
Syntax : DisplayRGB CopList#,Register,line,r,g,b[,copoffset] ;ecs=1 aga=4
```

1.15 DISPLAYLIB

```
Statement: DisplayUser
-------------------------------------------------------------------------------
Modes  :
Syntax : DisplayUser CopList#,Line,String[,Offset]
```

1.16 DISPLAYLIB

```
Statement: DisplayScroll
-------------------------------------------------------------------------------
Modes  :
Syntax : DisplayScroll CopList#,&xpos.q(n),&xpos.q(n)[,Offset]   ;3
```

DisplayScroll allows the program to dynamically display any part of a
bitmap on any line of the display. DisplayScroll should always follow
the DisplayBitMap command. The parameters are two arrays holding a list
of xoffsets that represent the difference in horizontal position from
the line above. AGA machines are able to use the fractional part of
each entry for super hiresolution positioning of the bitmap. Three
instructions per line are required for the DisplayScroll
command.

1.17 DISPLAYLIB

```
Statement: CustomSprites
-------------------------------------------------------------------------------
Modes  :
Syntax : CustomSprites CopList#,CCOffset,YPos,NumSprites ;4n+2
```

CustomSprites inserts a copper list that reinitialises the sprites
hardware at a certain vertical position in the display. These lower
sprites are assigned sprite numbers of 8..15.

1.18 DISPLAYLIB

```
Statement: CustomChunky
-------------------------------------------------------------------------------
Modes  :
Syntax : CustomChunky CopList#,CCOffset,YPos,width,height,res ;4n+2
```

1.19 DISPLAYLIB

```
Statement: CustomChunkySize
--------------------------------------------------------------------------------
Modes  :
Syntax : CustomChunkySize width,height
```

1.20 DISPLAYLIB

```
              .-------------------------------------------------------------------
|                                    DISPLAYLIB                                 |
`-----------------------------------------------------------------------------'
```

            Overview

                                        Command Index

         CopList

         CreateDisplay

         CustomChunky

         CustomChunkySize

         CustomColors

         CustomSprites

         CustomString

         DisplayAdjust

         DisplayBitMap

         DisplayControls

         DisplayDblScan

         DisplayPalette

         DisplayRainbow

         DisplayRGB

         DisplayScroll

```
            DisplaySprite

            DisplayUser

            InitCopList
```