BLITZCOPLIB

Conversion program

COLLABORATORS

	TITLE : BLITZCOPLIB	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Conversion program	October 9, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

BLITZCOPLIB

1.1 Overview of BLITZCOPLIB

Overview

An Acid Software Library

Converted to AmigaGuide by

Red When Excited Ltd

Used with the permission of Acid Software

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

1.2 BLITZCOPLIB

```
Statement: Slice
------------------------------------------------------------------------------
Modes  : Amiga/Blitz
Syntax : Slice [Slice#,Y,Flags]|[Slice#,Y,Width,Hite,Flgs,Bitplns,Sprts,Colrs, ↩
    Fscrnw,Bscrnw]

Slice Slice#,Y,Width,Height,Flags,BitPlanes,Sprites,Colours,Width1,Width2

The Slice command is used to create a Blitz 2 slice object. Slices are
primarily of use in Blitz mode, allowing you to create highly
customized displays.

In both forms of the Slice command, the Y parameter specifies the
'downwards' pixel position of the top of the slice. A Y value of 44
will position slices at about the top of the display.

In the first form of the Slice command, Flags refers to the number of
```

bitplanes in any bitmaps (the bitmap's depth) to be shown in the slice. This form of the Slice command will normally create a lo-res slice, however this may be changed to a hi-res slice by adding eight to the Flags parameter. For instance, a Flags value of four will set up a lo-res, 4 bitplane (16 colour) slice, whereas a Flags value of ten will set up a hi-res, 2 bitplane (4 colour) slice. The width of a slice set up in this way will be 320 pixels for a lo-res slice, or 640 pixels for a hi-res slice. The height of a slice set up using this syntax will be 200 pixels on an NTSC Amiga, or 256 pixels on a PAL Amiga.

The second form of the Slice command is far more versatile, albeit a little more complex.

Width and Height allow you to use specific values for the slice's dimensions. These parameters are specified in pixel amounts.

BitPlanes refers to the depth of any bitmaps you will be showing in this slice.

Sprites refers to how many sprite channels should be available in this slice. Each slice may have up to eight sprite channels, allowing sprites to be 'multiplexed'. This is one way to overcome the Amiga's 'eight sprite limit'. It is recommended that the top-most slice be created with all eight sprite channels, as this will prevent sprite flicker caused by unused sprites.

Colours refers to how many colour palette entries should be available for this slice, and should not be greater than 32.

Width1 and Width2 specify the width, in pixels, of any bitmaps to be shown in this slice. If a slice is set up to be a dual-playfield slice, Width1 refers to the width of the 'foreground' bitmap, and Width2 refers to the width of the 'background' bitmap. If a slice is NOT set up to be a dual-playfield slice, both Width1 and Width2 should be set to the same value. These parameters allow you to show bitmaps which are wider than the slice, introducing the ability to smooth scroll through large bitmaps.

The Flags parameter has been left to last because it is the most complex. Flags allows you control over many aspects of the slices appearance, and just what effect the slice has. Here are some example settings for Flags:

Flags setting	Effect	Max BitPlanes
$fff8	A standard lo-res slice	6
$fff9	A standard hi-res slice	4
$fffa	A lo-res, dual-playfield slice	6
$fffb	A hi-res, dual-playfiled slice	4
$fffc	A HAM slice	6 only

WARNING - the next bit is definitely for the more advanced users out there! Knowledge of the following is NOT necessary to make good use of slices.

Flags is actually a collection of individual bit-flags. The bit-flags

control how the slices 'copper list' is created. Here is a list of the bits numbers and their effect:

```
Bit #  Effect

15     Create copper MOVE BPLCON0
14     Create copper MOVE BPLCON1
13     Create copper MOVE BPLCON2
12     Create copper MOVE DIWSTRT and MOVE DIWSTOP
10     Create copper MOVE DDFSTRT and MOVE DDFSTOP
8      Create copper MOVE BPL1MOD
7      Create copper MOVE BPL2MOD
4      Create a 2 line 'blank' above top of slice
3      Allow for smooth horizontal scrolling
2      HAM slice
1      Dual-playfield slice
0      Hi-res slice - default is lo-res
```

Clever selection of these bits allows you to create 'minimal' slices which may only affect specific system registers.

The BitPlanes parameter may also be modified to specify 'odd only' or 'even only' bitplanes. This is of use when using dual playfield displays, as it allowins you to create a mid display slice which may show a different foreground or background bitmap leaving the other intact. To specify creation of foreground bitplanes only, simply set bit 15 of the BitPlanes parameter. To specify creation of background bitplanes only, set bit 14 of the BitPlanes parameter.

1.3 BLITZCOPLIB

```
Statement: CopLoc
------------------------------------------------------------------------------
Modes  :
Syntax : CopLoc
```

CopLoc returns the memory address of the Blitz mode copper list. All Slices, ColSplits, and CustomCops executed are merged into a single copper list, the address of which may found using the CopLoc function.

1.4 BLITZCOPLIB

```
Statement: CustomCop
------------------------------------------------------------------------------
Modes  :
Syntax : CustomCop Copins$,Y
```

CustomCop allows advanced programmers to introduce their own copper instructions at a specified position down the display. Copins$ refers

to a string of characters equivalent to a series of copper
instructions. Y refers to a position down the display.

1.5 BLITZCOPLIB

```
Statement: Show
------------------------------------------------------------------------------
Modes  :
Syntax : Show Bitmap#[,X,Y]
```

Show is used to display a bitmap in the currently used slice. This slice should not be a dual-playfield type slice. Optional X and Y parameters may be used to position the bitmap at a point other than it's top-left. This is normally only of use in cases where a bitmap larger than the slice width and/or height has been set up.

1.6 BLITZCOPLIB

```
Statement: ShowF
------------------------------------------------------------------------------
Modes  :
Syntax : ShowF Bitmap#[,X,Y[,ShowB X]]
```

ShowF is used to display a bitmap in the foreground of the currently used slice. The slice must have been created with the appropriate Flags parameter in order to support dual-playfield display.

Optional X and Y parameters may be used to show the bitmap at a point other than it's top-left. Omitting the X and Y parameters is identical to supplying X and Y values of 0.

The optional ShowB x parameter is only of use in special situations where a dual-playfield slice has been created to display ONLY a foreground bitmap. In this case, the X offset of the background bitmap should be specified in the ShowB x parameter.

1.7 BLITZCOPLIB

```
Statement: ShowB
------------------------------------------------------------------------------
Modes  :
Syntax : ShowB Bitmap#[,X,Y[,ShowF X]]
```

ShowB is used to display a bitmap in the background of the currently used slice. The slice must have been created with the appropriate Flags

parameter in order to support dual-playfield display.

Optional X and Y parameters may be used to show the bitmap at a point other than it's top-left. Omitting the X and Y parameters is identical to supplying X and Y values of 0.

The optional ShowF x parameter is only of use in special situations where a dual-playfield slice has been created to display ONLY a background bitmap. In this case, the X offset of the foreground bitmap should be specified in the ShowF x parameter.

1.8 BLITZCOPLIB

Statement: CopLen

Modes :
Syntax : CopLen

CopLen returns the length, in bytes, of the Blitz mode copper list. All Slices, ColSplits, and CustomCops executed are merged into a single copper list, the length of which may found using the CopLen function.

1.9 BLITZCOPLIB

Statement: ColSplit

Modes :
Syntax : ColSplit Colour Register,Red,Green,Blue,Y

ColSplit allows you to change any of the palette colour registers at a position relative to the top of the currently used slice. This allows you to 're-use' colour registers at different positions down the screen to display different colours.

Y specifies a vertical offset from the top of the currently used slice.

1.10 BLITZCOPLIB

Statement: ShowBlitz

Modes :
Syntax : ShowBlitz

ShowBlitz redisplays the entire set up of slices. This may be necessary if you have made a quick trip into Amiga mode, and wish to return to

```
Blitz mode with previously created slices intact.
```

1.11 BLITZCOPLIB

```
Statement: FreeSlices
------------------------------------------------------------------------------
Modes  :
Syntax : FreeSlices

FreeSlices is used to completely free all slices currently in use. As
there is no capability to Free individual slices, this is the only
means by which slices may be deleted.
```

1.12 BLITZCOPLIB

```
Statement: DisplayOff
------------------------------------------------------------------------------
Modes  :
Syntax : DisplayOff
```

1.13 BLITZCOPLIB

```
Statement: DisplayOn
------------------------------------------------------------------------------
Modes  :
Syntax : DisplayOn
```

1.14 BLITZCOPLIB

```
Statement: SetBPLCON0
------------------------------------------------------------------------------
Modes  :
Syntax : SetBPLCON0 BPLCON0 Default

bit#1-ERSY external sync (for genlock enabling)
bit#2-LACE interlace mode
bit#3-LPEN light pen enable
```

1.15 BLITZCOPLIB

```
Statement: SetBLWait
------------------------------------------------------------------------------
Modes  :
Syntax : SetBLWait Wait move for line 256 (eg:$ffe1fffe
```

1.16 BLITZCOPLIB

```
                .-------------------------------------------------------------------------
|                                 BLITZCOPLIB                                  |
`------------------------------------------------------------------------------'
```

Overview

Command Index

ColSplit

CopLen

CopLoc

CustomCop

DisplayOff

DisplayOn

FreeSlices

SetBLWait

SetBPLCON0

Show

ShowB

ShowBlitz

ShowF

Slice