

# **Plotter3D**

Sven Steiniger

<b>COLLABORATORS</b>
----------------------

	<i>TITLE :</i> Plotter3D		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Sven Steiniger	May 28, 2025	

<b>REVISION HISTORY</b>
-------------------------

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>Plotter3D</b>	<b>1</b>
1.1	main . . . . .	1
1.2	node_intro . . . . .	1
1.3	node_install . . . . .	2
1.4	node_system . . . . .	2
1.5	node_usage . . . . .	2
1.6	node_author . . . . .	3
1.7	node_localization . . . . .	3
1.8	node_disclaimer . . . . .	3
1.9	node_mui . . . . .	4
1.10	node_distribution . . . . .	4
1.11	node_thanks . . . . .	5
1.12	node_tooltypes . . . . .	5
1.13	node_tools . . . . .	6
1.14	node_index . . . . .	9
1.15	node_menu . . . . .	10
1.16	menu_project_about . . . . .	10
1.17	menu_project_aboutmui . . . . .	10
1.18	menu_project_survivor . . . . .	11
1.19	menu_project_abortquit . . . . .	11
1.20	menu_project_quit . . . . .	11
1.21	menu_settings_mui . . . . .	11
1.22	node_extrapages . . . . .	11
1.23	node_window . . . . .	12
1.24	window_settings_stdbuttons . . . . .	12
1.25	window_settings_stdmenu . . . . .	13
1.26	window_settings_global . . . . .	13
1.27	window_settings_imagebuttons . . . . .	14
1.28	window_settings_preview . . . . .	15
1.29	window_settings_buffer . . . . .	15

---

1.30	window_settings_realtime . . . . .	16
1.31	window_control . . . . .	19
1.32	window_functiontype . . . . .	20
1.33	window_editfunction . . . . .	21
1.34	window_preview . . . . .	22
1.35	window_transformation . . . . .	24
1.36	window_complexpreview . . . . .	26
1.37	window_light . . . . .	26
1.38	window_editlightsource . . . . .	27
1.39	window_calc . . . . .	28
1.40	window_material . . . . .	30
1.41	window_realtime . . . . .	31
1.42	window_clipboard . . . . .	33
1.43	window_informations . . . . .	33
1.44	window_export . . . . .	33
1.45	window_lettering . . . . .	34
1.46	group_edit3dfunction . . . . .	35
1.47	group_editmandelfunction . . . . .	36
1.48	group_editcsvfunction . . . . .	37
1.49	group_editobjectfunction . . . . .	38
1.50	group_editimagefunction . . . . .	39
1.51	node_functioncreating . . . . .	40
1.52	node_formula . . . . .	41
1.53	node_previewtypes . . . . .	42
1.54	node_boundingbox . . . . .	43
1.55	node_lightsources . . . . .	43
1.56	node_colorcontrol . . . . .	44
1.57	node_savefiles . . . . .	45
1.58	node_dragdrop . . . . .	45
1.59	node_problems . . . . .	46
1.60	gadget_color . . . . .	47
1.61	gadget_numerical . . . . .	47
1.62	gadget_font . . . . .	48

# Chapter 1

## Plotter3D

### 1.1 main

Plotter3D  
Version 3.35

Copyright 1996-98 by Sven Steiniger

Introduction	What is Plotter3D?
Installation	No Problemo.
System requirements	What's necessary?
Usage	Description.
Tooltypes	Tooltypes
Autor	Email etc.
Disclaimer	
Distribution	Freeware.
Credits	What was used.
History	What has changed.
Localization	
About MUI	The GUI.
Extra Tools	Plotter3D:Tools/
Problems	Bug or not?
Index	from A to Z

### 1.2 node\_intro

Introduction

---

Plotter3D was developed to draw three-dimensional functions.  
Therefore its possible to modify functions interactivly and render  
high quality pictures (Modies).

---

Besides Plotter3D has a special Realtime mode where the function is lighted by a lightsources and you can rotate, scale and move the function in realtime.  
Now Plotter3D is no longer limited to functions. It can also use CSV-files, TDDD-objects (Imagine), pictures and fonts.

Supporting graphic-cards (CyberGfx-System) was a main goal during development. Therefore its no problem to draw the preview without any lost of quality direct in 24Bit on a CyberGfx-Screen.  
Plotter3D still runs without any problems on normal Amigas except that some features need at least Kick V39.

The most Windows of Plotter3D are interactiv, means whenever you change values they are used immediately. Nevertheless you can undo the settings anytime. Its no problem to change the color of a lightsource, draw a preview of the result and you can reject the settings, because the old color looked better.

Besides special features of MUI like BubbleHelp, Drag&Drop, dynamic objects (use as many windows as you want) and online help are used.

## 1.3 node\_install

Installation

---

Its recommended to use the Install program that comes with the Plotter3D-package.

To do the installation by hand you simply copy the Plotter3D-directory to an appropriate place and make an assign Plotter3D: to it.  
(assign Plotter3D: "yourpath/Plotter3D")

## 1.4 node\_system

System requirements

---

Plotter3D needs:

- Kick V36, recommended V39 or above
- MUI3.8
- for realtime drawing a processor 68020 or above

## 1.5 node\_usage

Usage

---

If you use Plotter3D the first time you should first enter your Monitoraspect (menu Settings/Global...). Now you can load some functions (Control-window/Load) and experiment in the Preview-window. Besides you should have look at Settings-windows  
Global  
Preview.

Plotter3D features an online-help. Means if you want eq. an explanation of the Material window simply activate the window and press HELP-key.

You can find a description of the menu here.  
And a list of the different windowtypes exists here.  
All other pages can be found here.

## 1.6 node\_author

Author

---

Send bugreports, ideas, translations :), comments, nice functions, etc. to

Sven Steiniger  
Email: [ss37@inf.tu-dresden.de](mailto:ss37@inf.tu-dresden.de)  
Home : <http://www.inf.tu-dresden.de/~ss37>

You can download actual beta-versions from my homepage.

## 1.7 node\_localization

Localization

---

As english is not my native language you will find a lot of mistakes in this documentation or the catalog. Sorry.

Send me an email if you wanna make a translation into correct english or into another language.

## 1.8 node\_disclaimer

Disclaimer

---

I (the author) am not responsible for any damage caused directly or indirectly by this software package.

---

## 1.9 node\_mui

MUI

---

---

This application uses

MUI - MagicUserInterface

(c) Copyright 1993-96 by Stefan Stuntz

MUI is a system to generate and maintain graphical user interfaces. With the aid of a preferences program, the user of an application has the ability to customize the outfit according to his personal taste.

MUI is distributed as shareware. To obtain a complete package containing lots of examples and more information about registration please look for a file called "muiXXusr.lha" (XX means the latest version number) on your local bulletin boards or on public domain disks.

If you want to register directly, feel free to send

DM 30.- or US\$ 20.-

to

Stefan Stuntz  
Eduard-Spranger-Straße 7  
80935 München  
GERMANY

Support and online registration is available at

<http://www.sasg.com/>

## 1.10 node\_distribution

Distribution

---

---

Plotter3D is Freeware.

Means it can be freely distributed as long as no file is modified and all files are kept together. Besides its not allowed to make any Profit with

---

this package.

## 1.11 node\_thanks

Credits

---

The following programs were used during the development of Plotter3D:

- o AmigaE, an excellent compiler from Wouter van Oortmerssen
- o GoldED, the editor from Dietmar Eilert
- o MUI, the GUI written by Stefan Stuntz
- o FlexCat from Jochen Wiedmann
- o Enforcer from Michael Sinz
- o Sushi from Carolyn Scheppner
  - o Autobuild from Jan Hendrik Schulz

Thanks go to:

- o Jerome Chesnot for french translation and for being the best translator. He does translation faster than I can add new text to Plotter3D. Contact him via [jchesnot@noname.fr](mailto:jchesnot@noname.fr).
- o Toño Diaz for spanish translation. Because of several problems (HD-crash) he needs 11 months to start translation but then did it within one day =). Contact via [parsec@eyesoftime.com](mailto:parsec@eyesoftime.com).
- o Jason R. Hulance for his many Aminet-uploads for AmigaE
- o Markus "Küßchen" Seifert for letting me use his computer for all the (cybergfx) testing.
- o All the people who send me emails no matter if they reported bugs or just send flames (of course I really love the second ones =).

## 1.12 node\_tooltypes

Tooltypes

---

All options that can't be changed at running-time are changed via tooltypes. Their main usage is to switch off some features of Plotter3D for testing. Therefore think twice before you change them. Supported tooltypes are:

UseLocale

If it exists, the localization is switched on.

UseCyberGfx

Put it into brackets to disable cybergfx-support.

---

UseCyberGL

Not used yet.

UseC2PHardware

Influence the selection of c2p-routines both for rendering and realtime mode.

It's main goal is as a workaround for possible bugs in c2p-routines (which had been occurred some time ago).

Values:

- 0 - check if c2p hardware is installed (default)
- 1 - force to use systems c2p-routines  
(WriteChunkyPixels()). Useful if you eq. have special patches installed.
- 2 - never use c2p hardware.

## 1.13 node\_tools

Extra tools

---

Tools/Analyse\_SaveFile

Shows the contents of an file created by Plotter3D.

Tools/PalettesMaster

With this tools you can create new palettes for realtime mode.

Plotter3D uses a special format for these palette files which enables simple and fast loading.

Usage:

Load

Filename

Select the popup-button and select a palette to load or just enter an filename.

The palette is then loaded automatical.

Known types are all chunky-palettes and standard ilbm-palettes.

Reload

Reloads the selected file.

Edit

Palette name

Nick name of the loaded palette

Colors

Number of colors

New

Creates an new palette.

Use the popup-button to specify the number of colors and how the palette is initialized.

For initialisation the standard phong-illumination model is used.

---

#### Edit colors

Opens MUI's palette requester. Just select a color and modify it.

#### Show palette

Opens an screen and displays the palette.  
Press LMB to close the screen.

#### Show and edit palette

Opens an screen and displays the palette.  
It also opens the edit-palette-requester from 'regtools.library' where you can modify the palette.

#### Save

##### Filename

Where to save the palette.

##### Type

Select the type of the saved palette.  
You should select 'Chunky' or 'ILBM'.

Other types are:

##### LoadRGB

special Plotter3D-palette-file (always ends on '.pal256') that contains all datas needed for an LoadRGB32()-call.

##### RAW24

Such a file has 3x8Bit (red,green,blue) for every palette entry.

#### Save

Save the palette.

#### Quit

Quits PalettesMaster.

#### Notes on palettes:

- the palette must contain 256 colors
- color 0 is background color. Means it should be black.  
Press "Set color 0 to black" if you are not sure if this is the case.
- color 1 should have lowest intensity and color 255 should have highest intensity

#### Tools/ChunkyMaster

Converts an picture into chunky-datas which can be used as texture or faked-phong texture for Plotter3D's realtime mode.

Realtime mode also accepts other pictures but loading of chunky-pictures is much faster and uses a way less memory.

Usage:

##### Load

Filename

Press the popup-button to select an picture or just enter an filename.  
It accepts chunky-pictures, ilbm-pictures and pictures of any types that are supported by 'datatypes.library'. No restrictions in colorcount or resolution exists.

#### Reload

Reload the picture.

#### Edit

##### Info

Picture name

Nick name of the loaded picture.

Width

Width of the picture.

Height

Height of the picture.

Colors

Number of colors.

Shadetable

Has the picture an shadetable?

Show picture

Opens an screen and displays the picture.

##### Scale

Scale picture

Scales the loaded picture to any resolution. It uses the contents of the 'New width' and 'New height' gadgets as width and height.

##### Shadetable

Reduce colors to

At first the used colors of image are reduced to this value. The rest is filled with darker/lighter ones.

Shade steps

Number of shade values. Plotter3D uses 64.

Minimal intensity

The intensity of the darkest shade value. Normally 0% which means full black.

Maximal intensity

The intensity of the brightest shade value. A value of 100% displays the

normal picture, higher values make them more shiny.

Calculate

Calculates the shadetable. This may take some time.

Show shadetable

Shows the calculated shadetable. It fades the picture from darkest to brightest shade value.

Save

Filename

Where to save the picture.

Type

The type of the picture.

Save shadetable

If selected then the shadetable is saved within des picture.

Pack XPK

If selected, the picture-datas are packed with help of 'xpkmaster.library' (Only 'chunky'-images).

Packer

The nick name of the xpk-packer. Recommend are "SMPL" (fast, good compression) and "LZCB" (slow, excellent compression).

Save

Save the picture.

Quit

Quits ChunkyMaster

Notes:

Plotter3D uses pictures with size 256x256. It can use smaller or larger pictures but this images must then be scaled which requires memory and cpu-time. Therefore scale the image with an image processing program (like ImageFX or AdPro) or use ChunkyMaster's scale option.

## 1.14 node\_index

Index

---



---

## 1.15 node\_menu

Menu

---



---

Project

About...

About MUI...

Amiga survivor...

-----

Abort & Quit

Quit

Settings

MUI...

-----

Global...

Imagebuttons...

Preview...

Realtime...

Windows

Render...

Realtime...

Information...

Export...

-----

Complex preview...

Transformation...

-----

D&D Clipboard...

## 1.16 menu\_project\_about

Project/About...

---



---

Opens a window with information about Plotter3D.

The listview contains actual information about the system and the internal datas of Plotter3D.

See also: Credits

## 1.17 menu\_project\_aboutmui

Project/About MUI...

---



---

Opens the standard MUI-About-window.

---

## 1.18 menu\_project\_survivor

Project/Amiga survivor...

---

Opens the survivor window.

## 1.19 menu\_project\_abortquit

Project/Abort & Quit

---

Quits Plotter3D without any safety-requesters. All functions are deleted.

## 1.20 menu\_project\_quit

Project/Quit

---

Same function as Quit-button in the Control-window.  
Quits Plotter3D, for every unsaved function a safety-requester is shown.

## 1.21 menu\_settings\_mui

Settings/MUI...

---

Opens the MUI-Configuration-Window for Plotter3D.

## 1.22 node\_extrapages

Other pages

---

Drag&Drop  
Edit 3D-Function  
Edit Mandel-Function  
Edit CSV-Function  
Edit Object-Function  
Edit Image-Function  
Function creation  
Allowed formulas

---

```

Previewtypes
BoundingBox
Lightsources
Colorselection
SaveFiles
Color gadget
Numerical gadgets
Font gadget

```

## 1.23 node\_window

Windows

---



---

```

Complex Preview
Control
D&D Clipboard
Edit function
Edit lightsource
Export
Functionstype
    Information
Lettering
Lighting
Material
Preview
Realtime
Render
Settings - Buffer
Settings - Global
Settings - Imagebuttons
Settings - Preview
Settings - Realtime
Transformation

```

## 1.24 window\_settings\_stdbuttons

Standard Buttons

---



---

Cancel

Closes the Prefs-Window. The settings are not changed.

Use

Closes the Prefs-Window. The actual settings are used by now.

Save

Like Use-Button but the settings are also saved in the directory Plotter3D:Prefs/ and are used everytime you run

---

Plotter3D.

## 1.25 window\_settings\_stdmenu

PopUpMenu

---

Plotter3D uses Object-specific Popupmenus. More Information on this topic can be found within the 'Readme'-file of MUI.

Defaults/Reset to defaults

Resets the settings to internal default values.

Defaults/Last saved

Resets the settings to last saved values.

Default/Restore

Restore the settings to last used values.

## 1.26 window\_settings\_global

Global Settings

---

General

Monitor aspect

Simply enter the ratio of monitor width to the height.

Doublebuffering

Determines how the preview is shown. If you draw directly into the window you see a strong flickering. Therefore Plotter3D has several methods to avoid this:

None

Switches buffering off. Because of the speed of graphic cards there should be not too much flickering. Using this method on normal Amigas is nearly impossible.

Singeplane

Plotter3D uses a bitplane as buffer. This method is very fast and avoids flickering. On graphic cards (with chunky-mode) it may slow down the drawing.

Full

Plotter3D uses a bitmap with the same depth as the screen.

---

#### Cyclegroup level

Whenever an registergroup has not less than  
 'cyclegroup level'-entries a group with an cyclegadget  
 is shown (saves a lot of space).  
 Set it to 1 to always use cyclegroups or to 10 to  
 always use registergroups.

#### Always use colorwheel

Normally Plotter3D only use colorwheels on High/True  
 color screens. With this witch they are used always.

#### Default paths

Sets default paths for functions, materials, lightsources and  
 exported objects. Use popup button to use a ASL-filerequester  
 for path selection.

#### Default function settings

If you try to render an complex preview which needs an  
 lightsource but the functions doesn't have one, the lightsource  
 and material you specified here are use temporarily.

See also: Cancel/Use/Save

PopUpMenu

Drag & Drop

Numerical gadgets

## 1.27 window\_settings\_imagebuttons

Settings for Imagebuttons

---

#### Toolbar

Here you define the Imagebuttons used by Plotter3D. To change  
 the image of an button simply select the entry in the listview  
 and enter a filename into the textgadget. You can also use an  
 ASL-Requester if you press PopUp-Button beside the textgadget.  
 The selected image is shown above the textgadget or in the  
 case of an error a '??'.  
 You can choose ILM-brush and ILM-pictures as images. But it  
 should not be too large because MUI may have problems to  
 reopen the Preview window.  
 The standard size is 26x26 pixel.

#### Button

Here you can choose the images for the decrease/increase  
 buttons beside all numerical gadgets.

See also: Cancel/Use/Save

PopUpMenu

Drag & Drop

---

## 1.28 window\_settings\_preview

Settings for Preview

---

### Flags

#### Fast preview

If selected Plotter3D draws only a boundingbox during rotation/moveing a function with the mouse.

#### Draw boundingbox

Draws a boundingbox around the function.

#### Draw lightsources

Also draws all lightsources fo the function.

#### Update lightsources

If enabled and you change a lightsource the preview is updated automatical.

### Buffer

See Buffersettings.

See also: Cancel/Use/Save

PopUpMenu

## 1.29 window\_settings\_buffer

Settings for rendering buffer

---

### General

#### Bufferheight

Plotter3D uses a depth buffer and a color buffer to render the pictures (all complex calculations except Monochrom). The width of these buffers always correspond to the width of the picture. Holding the whole buffer (5\*width\*height bytes) requires a lot of memory. Therefore Plotter3D renders only parts of the picture. You can define the height of one part with the slider. Of course rendering many parts slows down the calculating that's why you should not enter values lower than 50 pixels.

#### Dithermode

Plotter3D calculates always with 24Bit

---

(16.8 million colors). Because this number of colors can only be shown on graphic cards it must be reduced. With this option you can define the used method.  
See also Colorselection of Plotter3D.

#### Save buffer

The calculated Image can be saved automatical.

#### Always

Switch saving on/off.

#### File type

Defines the type of the saved file.

#### Filename

Simply enter the name of the file. You can also use an ASL-Requester with the Popupbutton.

#### Also save function

If enabled the function is also saved. This may be as an seperate file or included into the picture-file (IFF).

#### Fonts

Defines the fonts used for lettering the image.  
See also Lettering.

#### Title

The font used for the imagetitle.  
An Fontgadget is used for this.

#### Author

The font used for author notice.

See also: Cancel/Use/Save

PopUpMenu

Drag & Drop

Numerical gadgets

## 1.30 window\_settings\_realtime

Settings for Realtime mode

---



---

#### Type

##### Gouraud shading

Switches to gouraudshading. The color for every point is calculated and then spread continiously

---

on the area.

#### Palette

Select the used colorpalette. The palettes are read from 'Plotter3D:Palettes/'.

#### Texture mapping

Switches to texturemapping. An image (texture) is mapped on the object.

#### Texture name

Select the used texture. You can load ilbm-pictures, chunky-pictures or pictures supported by 'datatypes.library' of any resolution. But it's recommended to use chunky-pictures of size 256x256 because they can be loaded fast and with very few memory usage. You can create such textures with Tools/ChunkyMaster.

#### Faked phong

Switches to faked phongshading. It calculates the normales for every point and uses the 2 angles as coordinates into the texture. The appropriate texture part is then mapped on the area. The used image should be contain somewhat like an circle.

#### Faked phong texture

Select the used texture. You can load ilbm-pictures, chunky-pictures or pictures supported by 'datatypes.library' of any resolution. But it's recommended to use chunky-pictures of size 256x256 because they can be loaded fast and with very few memory usage. You can create such textures with Tools/ChunkyMaster.

#### Flat shading

Switches to flatshading. Every area is filled with the calculated color. Doesn't look nice with most objects but with many small areas it is nearly as good as gouraudshading but a lot faster.

#### Palette

Select the used colorpalette. The palettes are read from 'Plotter3D:Palettes/'.

#### Gouraud texture

Switches to combined gouraud shading/texture mapping. This requires an extra shading table which may either be already contained in the chunky image or must be calculated during precalculations (slow). You can create this shadetable with Tools/ChunkyMaster.

#### Gouraud texture name

Select the used texture. You can load ilbm-pictures, chunky-pictures or pictures

supported by 'datatypes.library' of any resolution. But it's recommended to use chunky-pictures of size 256x256 because they can be loaded fast and with very few memory usage. You can create such textures with Tools/ChunkyMaster.

#### Shade mode

Selects the initial shade mode. It can also be changed at running time via key 'm'.

#### Lighted (Dynamic)

The new color values are calculated for every frame depending on the normals of every area.

#### X coordinates

The color values depend on the x coordinates of the points.

#### Y coordinates

The color values depend on the y coordinates of the points.

#### Z coordinates

The color values depend on the z coordinates of the points.

#### Normals (Static)

Like 'Lighted' but is calculated only once.

#### Background picture

If selected the texture (if the realtime mode has one) is also used as background picture.

#### PalettesMaster

Runs external program 'PalettesMaster' where you can change or create Palettes.

#### ChunkyMaster

Runs external program 'ChunkyMaster' where you can create texturefiles from pictures.

### Screen

#### Screenmode

Here you select the screenmode. Press 'Select'-button and use the ASL-Requester to select an screen.

Note: There are some restrictions on AGA-computers. See Realtime mode for more information.

#### Double buffering

Selected this to avoid flickering. Needs more display memory.

Note: If you got an "Could not open screen" error, you should switch this option off.

#### Animation

Here you can set the format and the base name for the animation that is created during realtime mode.

Note: ILBM-files are currently saved as unpacked iffs.

For every frame a numerical prefix is added to basename. This makes it very easy to create and complete animation with eq. MainActor.

#### Rotation

Here you can set all initial values for the realtime mode.

##### Angles

Defines the rotation-angles around all axes.

##### Delta

This values are added to rotation-angles every frame.

##### Scale

The global scale-value.

#### Flags

See Realtime mode/Keyboard for further explanation.

#### i-Glasses

Here all settings for i-Glasses mode are made.

##### Use i-glasses

Switches i-Glasses mode on/off.

##### Eye distance

Defines the virtual eye distance.

See also: Cancel/Use/Save

PopUpMenu

Drag & Drop

Numerical gadgets

## 1.31 window\_control

#### Control-Window

---



---

This is the main window of Plotter3D.

#### Functions

Contains a list of all functions and some buttons. The selected function is shown in the Preview-window.

If you click twice on a function the corresponding Edit-window is opened.

If you press the right mousebutton above the list you got an popupmenu where you can open special editwindows for the

---

selected function:  
 Lightsources...  
 Opens a Lighting-Window.

Material...  
 Opens a Material-Window.

Lettering...  
 Opens a Lettering-Window.

Edit...  
 Opens a Edit-Window.

The list consists of 2 columns: the first shows the name of the function and the second the type.

New  
 Creates a new function. A Functiontype-window is opened to select the type of the function.

Delete  
 Deletes the selected function. If these functions is modified a safety-requester is shown.

Load  
 Opens a ASL-Requester where you can select the function(s) to be loaded.  
 See also Plotter3D Savefiles.

Save  
 Opens a ASL-Requester where you can enter the filename of this function.  
 Note:  
 Plotter3D saves all datas in these file  
 (function parameter, transformation, material  
 and lighting (if exists)).  
 See also Plotter3D Savefiles.

Quit  
 Quits Plotter3D. For every modified function a safety-requester is shown.

See also: Drag & Drop

## 1.32 window\_functiontype

Functionstype-Fenster

---

Here you determine the type of an new function.  
 After you have selected a type the Edit-window is opened.

3D-Function

---

Creates a 3D-function. You can enter equations for X-, Y-, Z-values with dependencies from parameters U and V.

#### Mandel

Creates a Mandelbrot-function. You can define values for real and imaginary axis and maximum iteration depth. See a good Fractal program (eq. Mandel2000) for an explanation on Mandelbrot fractals.

#### CSV

Creates a CSV-function (Comma Separated Values). In such a file every line represents one value group. You can enter the text file and how it is parsed.

#### Import Object

Imports a custom (3D-) object (eq. Imagine objects).

#### Import Picture

Imports a 2D-data file, usually a picture or a font.

#### Cancel

Closes the window without creating a new function.

## 1.33 window\_editfunction

### Edit Function

---

This window consists of two parts. The left one is equal for all functions. The contents of the right one depends on the function type. The type is also shown in the window title.

#### Left part

##### Name

The name of the function. The name is also shown in the function list of Control-window. Therefore it should be short and unique.

##### Mesh

Contains definition about the internal mesh. May be hidden if the function has no mesh.

##### x-Size

The horizontal mesh size. Normal values between 20 and 100.

##### y-Size

The vertical mesh size. Normal values between 20 and 100.

##### Need-Text

Shows the necessary memory (in bytes) for these functions.

---

#### Merge tolerance

May be hidden if the function has no merge option.

This controls the merging of points. All points that are within a sphere with radius 'merge tolerance' are merged together.

This removes ugly edges when function is used in realtime mode.

#### Check

Checks all values of these function. A window with the result is shown automatical.

#### Cancel

Closes the window. All settings remains unchanged.

#### Ok

Closes the window. The actual settings are used and necessary calculations are made.

A function check is preformed automatical. If an error occurs an error-requestor occurs and the edit window stays open.

#### Right part

Depends on the function type.

See: Edit 3D-Function

Edit Mandel-Function

Edit CSV-Function

Edit Object-Function

Edit Image-Function

See also: Drag & Drop

Numerical gadgets

## 1.34 window\_preview

Preview-window

---

This window is used for general function transformation and drawing a function. All actions change only the function which is selected in the Control-window.

The following keys are used:

7,8,9 - rotate around X-axis (backwards, fast forward, forward)

4,5,6 - rotate around Y-axis (backwards, fast forward, forward)

1,2,3 - rotate around Z-axis (backwards, fast forward, forward)

s,S - zoom in, out

x,X - move left, right

y,Y - move down, up

z,Z - move to back, to front

d - reset to default

Space - draw complex preview

---

The window consists of four parts. The middle part is the drawing area.

#### Left part

The description goes from top to bottom. The images of the buttons can be changed with Menu Settings/Imagebuttons.

##### 1. Button

Set mouse handling to rotation.

Means whenever you press the left mousebutton above the drawarea (and hold down the mousebutton) you can rotate the function around X- and Y-axis by moving the mouse. To rotate around Z-axis press right mousebutton.

##### 2. Button

Set mouse handling to movement.

With LMB you control movement along X- and Y-axis.

When pressing RMB you can change the zoom factor.

##### 3. Button

Switches into select mode.

By pressing any mousebutton you can select an point of the actual function. The coordinates of this point are displayed within the drawing area.

With this mode you can eg. roughly find minima/maxima of an function.

When the window is large enough same more imagebuttons may appear here. With these you can open Information, Render, Realtime window.

##### 4. Button

Opens the Transformation-window.

##### 5. Button

Opens the window for Complex preview.

#### Drawing area (middle part)

The preview of a function is drawn into this area. The preview type can be choosen with right part buttons.

If you press left or right mousebutton in this area you can change the transformation. You can select the mouse handling with the left part buttons.

#### Right part

Controls the preview type.

The description goes from top to bottom. The images of the buttons can be changed with Menu Settings/Imagebuttons.

##### 1. Button

Switchs preview off.

##### 2. Button

Set previewtype to simple preview. Only a boundingbox is draw.

---

### 3. Knopf

Set previewtype to point preview.  
Only the points of the functionmesh and a boundingbox (if selected in window Preview-settings) and/or the lightsources (window Preview-settings).

### 4. Button

Set previewtype to line-X preview.  
Draws lines alongs the X-axis between the points of the functionmesh. If selected also a draws a boundingbox and the lightsources.

### 5. Button

Sets previewtype to line-Z preview.  
Like line-X preview but connects the points along Z-axis.

### 6. Button

Combines line-X and line-Z preview.

### Lower part

These sliders enable you to rotate the function with a constant stepsize. With this you get an idea how the function looks.

Normally the knob stay in the middle of the slider. The further you move them to left/right the faster rotates the function backwards/forwards around an axis. If you release the knob it jumps back to zero position (middle).

#### RotX

Rotation around X-Axis

#### RotY

Rotation around Y-Axis

#### RotZ

Rotation around Z-Axis

## 1.35 window\_transformation

### Transformation-window

---



---

Here you control the transformation of the selected function. If you change a parameter a new preview is draw automatical in the Preview-window.

#### Rotate

rotation angle (degrees) around X-, Y-, Z-axis.

#### Scale

Scale value of X-, Y-, Z-axis in percent. 100% means no scale.

---

#### Move

Movement along X-, Y-, Z-axis.

#### Special

Settings you don't need very often.

#### Aspect ratio

With this you can define when aspect ratio and viewportsize correction is performed.

For explanations the following clipping cube is used:

X: -150..150; Y: -150..150; Z: -300..300 .

#### After clipping

This is the old method. First the planes are clipped against the clipping cube and then these cube is scaled into the viewport. Of it also takes care of the correct aspect ratio. Because of this you may see clipped planes within your viewport although there is enough room for drawing.

#### Before clipping

This method changes your clipping cube!

Imagine your viewport is 200 pixels width and 100 pixels height. Plotter3D then simply takes the Y-coordinates of your clipping cube and multiplies them by 2 (200/100) to get the X-coordinates of the clipping cube. Then all planes are clipped against the new clipping cube and this cube is then scaled into the viewport. You won't see any clipped planes within your viewport.

If your viewport is more heigher than widhter the Y-coordinates are calculated from the X ones.

Takes care of the correct aspect ratio.

#### Clipcube

All parts of function that are outside the cube are cutted. The cube is then projected into the drawing area.

You can enter the minimal/maximal values of the X-, Y-Z-coordinates. With the +/- Buttons the values can be change in steps of 0.5.

#### Zoom

Defines the zoom value in percent. 100% means original size.

#### R

Resets all parameters to internal default values.

See also: Drag & Drop

Numerical gadgets

## 1.36 window\_complexpreview

Complex Preview

---

Here you can select previewtypes that can not be drawn in realtime yet. The result is normally shown in Preview-window. (exception: when used with Render-window).

For an description of all previewtypes see Previewtypes. BubbleHelp is also available: If you place the mouse above the Cyclegadget and don't move the mouse a description of the selected previewtype is shown.

Draw Linemesh

If selected all areas are surrounded with a white line. This works not perfect because its really difficult to implement lines into z-Buffer. Even harder if also fixed point mathematics is used.

Line color

With this colorgadget you can choose the color of all lines.

Here you can select the type of colorclipping.

BubbleHelp is available.

Preserve colors

This is the old method. It first calculates all color values and then calculates an factor so that the brightest colorvalue lies within the valid colorcube if you multiply it with these factor. Then all colorvalues are multiplied with this factor. With does method you will lose brightness but the colors stay correct.

Crop colors

This is the same method as the one that phongshading uses. Here the colors are calculated and during drawing the colorvalue of every pixel is checked if it lies within the valid colorcube. If not the values (red, green, blue) are cropped to fit within the colorcube.

This method may change colors (eq. colorvalue 1.6,0.8,0.3 (red, green, blue component) is cropped to 1.0,0.8,0.3 which is of course a other color) but you get very nice pictures with bright shining reflections.

Draw

Draws the complex preview.

## 1.37 window\_light

Lighting-window

---

---

With this function you can control the lighting of an function. The name of the function is also shown in the windowtitle.

All changes are used immediately. Means you can change some parameter and look at the result in Preview-window. If you made an mistake you can reset to the old settings by pressing 'Cancel'.

#### Lightsources

This list contains the names, types and status of all lightsources.

If you doubleclick on the 'Status'-column the lightsource is toggled on/off. If you hit any other column the Edit-window of the lightsource is opened.

#### New

Creates a new lightsource. A window to select the type is opened.

#### Delete

Deletes the selected lightsource \*without\* safety-request.

#### Load

Loads new lightsources from a file. A ASL-requester is opened to select the file(s) to be loaded.

#### Save

Saves \*all\* lightsources into a file. If the file already exists a safety-regester is shown.

#### Cancel

Closes the window. The old settings are restored.

#### Ok

Closes the window and the settings are used.

See also: Drag & Drop

## 1.38 window\_editlightsource

### Edit Lightsource

---

With this window you can change the parameters of an lightsource. The name of the lightsource is also shown in the windowtitle.

All changes are used immediately. Means you can change some parameter and look at the result in Preview-window. If you made an mistake you can reset to the old settings by pressing 'Cancel'.

Edit lightsource

---

## Name

Defines the name of the lightsource. The name is shown in the lightsource list and when the lightsource is drawn. (Previewsettings).

## Color

Defines the color of the lightsource. A `colorgadget` is used to enter the values.

## Intensity

Defines the intensity of the lightsource. Means that the color values are multiplied by this number.

Local

If selected the lightsource is 'local'. Means it is transformed like the function.

## Status

Switches the lightsource on or off.

According to the lightsource type this part may contain further elements.

## Parallel Lightsource

## Direction

Defines the direction (vector) of the lightsource. With the floatgadgets you can enter the X-, Y-, Z-coordinates. With the +/- Buttons the coordinates can be changed in steps of 0.1.

## Point Lightsource

## Position

Defines the position of the lightsource.

Radius

Defines the maximum lighting radius. Every area outside an sphere with this radius are not lighted.

## Zoom radius

If selected the radius is scaled with the global zoom value.

See also: Drag & Drop

## Numerical gadgets

## Color gadget

### 1.39 window\_calc

Render-window

With this window the selected function is drawn. The name of the function is shown in the windowtitle.

The rendertype is always selected with the Complex Preview window. It is automatical opened when the Renderwindow is opened.

## Output

Plotter3D has 3 output modes.

### Output type

Defines the type of output.

#### Don't draw

Switch output off. Because no screen is opened it saves some memory. Of course without switching 'save on' (Buffersettings) this mode is senseless.

#### Custom screen

Plotter3D opens an own screen and draws the rendered picture on it.

#### Window on Publicscreen

Opens a window on a public screen. These option should only be used with 16/24Bit CyberGfx-screens.

## Output settings

Defines the parameter of the output types. Contains the page with settings for current output type.

### Picture

Settings for type 'Don't draw'.

#### Picturewidth

Width of the rendered picture.

#### Pictureheight

Height of the rendered picture.

### Custom screen

Settings for type 'Custom screen'.

#### Screenmode

Defines the screenmode. The textgadget contains Width x Height x Depth. With the 'Select...'-Button a Screenmode-requester is opened.

### Public screen

Settings for type 'Window on Publicscreen'.

#### Screenname

Defines the name of the public screen.

#### Window width

Width of the render window.

Window height  
Height of the render window.

Rendertype...

Opens the Complex Preview window to choose the rendertype.

Buffersettings...

Open the Buffersettings-window for rendering.

On this page you set things like type of color reducing or if the picture should be saved.

These settings only affect the renderingbuffer and are saved as seperate preference file.

Render

Starts the calculating.

Show Screen/Window

Pops the screen to front. A click on it or into the window puts the screen of Plotter3D back to front.

Close Screen/Window

Closes the screen or the window.

This is also done automatical if you leave the 'Render'-window.

See also: Drag & Drop

Numerical gadgets

## 1.40 window\_material

Material-window

---



---

In this window you can define the 'Material'-attributes of an function. These attributes are necessary for lighting the function. Therefore they are only used with Flat-, Gouraud- and Phongshading. This is also an interactiv window, means you can change the settings and test them (eq. with Complex preview). Nevertheless you can undo these settings with the 'Cancel'-button.

The sum of values for 'Ambient', 'Diffuse' and 'Specular' should be 1.0.

Ambient

Determines the share of ambient light in the color of an pixel. Normally values between 0 and 1. For special cases you can enter other values.

The red, green and blue shares of the lightsources are multiplied with this value.

Affects only ambient lightsources.

Diffuse

Like ambient but for diffuse share of the color.

---

Affects parallel lightsources.

Specular

Determines the share of high lights in the color of an pixel.  
The color depends *\*not\** from a lightsource.

## Spec. Power

Determines the size of the high lights. Small values results in big enlarged hight lights. Big values results in small sharp ones.

## Specular color

Determines the color of the high light. A `colorgadget` is used for entering the color values.

## Double normals

Normally a normal is calculated for every area. Because of this a may get ugly results after lighting since areas that are turned away appears black. If you switch on this option two opposite normals area used for every area and both sites of the areas are lighted equal.

## Load

Loads a new material definition.

Save

Saves the material definition.

Cancel

Closes the window. The old settings are used.

Ok

Closes the window. The new settings are used.

See also: Drag & Drop  
Numerical gadgets  
Color gadget

## 1.41 window realtime

Realtime-window

Plotter3D has an special realtime mode. With this its possible to view the function in realtime! The function can be rotated, moved and scaled.

```
*****
** These mode needs at least a processor of type 68020 and a      **
** graphic hardware that can display 256 colors (an AGA-chip or  **
** an CyberGfx-system)                                           **
*****
```

## Control

All supported keys are shown here.

## Change settings...

Opens the Realtiesettings-window.

## Go!

Have fun.

## Information

-----

How already described above you need at least an 68020 because special commands of this processor are used. But for real fun you need a fast 68030. On my computer (50Mhz 68030) and an functionmesh with the size of 20x20 (==800 areas) it runs in 12 pictures/s. Also FAST-Ram helps a lot (0.5 - 1 Mbyte).

If you own a graphic card you should choose an screenmode that runs on this card. This increases graphic speed and reduces necessary processor power. The picture datas must be converted on AGA-computers which costs about 33% of the processor power. Besides you should select always a LORES-screenmode because otherwise additional speed losses occur.

As all realtime routines are designed for speed, overflow-errors may easily occur when you zoom too deep into the object. If this happens, you will either see weird triangles drawn on your screen or a black screen labeled with 'Math overflow'. You should then reduce the zoom-value.

The drawing become faster if the backfaces are removed automatical (Shift h). This works very good for example with the "Torus"-function. In most cases this mode results in wrong (incomplete) pictures. In some cases these effects can be avoided by using option "inverse depthsorting" (i, Shift i) which sorts areas in inverse order.

With gouraudshading its recommended to switch "absolute normals" on. With this every face has two opposite normals and therefore both sides reflect the light.

The used palettes are loaded from 'Plotter3D:Palettes/'. Press key "c" to toggle to next palette. You can add own palettes with external program PalettesMaster.

By pressing left or right mousebutton you switch to mouse control mode. Now you can rotate the function around X- and Y-axis (LMB) and Z-axis (RMB). To switch back to normal mode press 'p'.

The frames can be saved as animation. Press 'Enter' to start recording (a counter is shown in the bottom left part). Press 'Enter' again to stop recording. The basename and the format of the animation is defined in Realtiesettings/Animation.

In i-Glasses mode you can change the eye distance with keys 'e' and 'E'. The current eye distance is displayed (with the actual rotation angles)

at the top left edge).

## 1.42 window\_clipboard

D&D Clipboard

---



---

Simply drag draggable objects into the list. In the same way you can drag entries from the list and drop them on objects which accept the drop (they are surrounded by a rectangle).

The contents of the list remain until you leave Plotter3D.

More information for Drag&Drop can be found within the MUI-documentation.

See Drag&Drop to get a clue on Plotter3D's drag&drop facilities.

Remove

Removes the selected entry.

Clear

Removes all entries.

## 1.43 window\_informations

Informations

---



---

Here you can find special information for the current function.

Boundingbox

Displays the maximum and minimum X-,Y-,Z-values.

Static

Displays static information like number of points and shapes.

Dynamic

Displays dynamical information like rotation and zoom values.

## 1.44 window\_export

Export

---



---

With this window you can export the current function.

Export settings

Type

Defines the type of the exportfile.

---

Supported formats are:

Imagine object

Creates an TDDD-IFF-file that can be used directly with Imagine.

CSV-File (Y)

Creates an CSV-file that contains the Y-values of the function.

CSV-File (X,Y,Z)

Creates an CSV-file that contains the X,Y,Z-values of the function seperated by commas.

Lightwave object

Creates an object file for Lightwave.

File

Defines the filename.

Note:

The standard filename extension is added automatical (if it isn't already there).  
(.IOB, .CSV, .LWO)

Export it!

Starts exporting. This can take a while.

See also: Drag & Drop

## 1.45 window\_lettering

Lettering

---

In this window you can define the lettering of an function. The lettering appears on images rendered with Complex preview or Render window.

This is also an interactiv window, means you can change the settings and test them (eq. with "Complex preview"). Nevertheless you can undo these settings with the 'Cancel'-button.

Here you only define the contents of the 2 letterings. All other settings like used fonts, colors etc. are defined via buffer preferences.

Title

Defines the title of this function. The 'Add'-button opens an popup showing all format codes. Simply double click an entry to insert the contents into the string gadget.

Supported format codes are:

- %FUNCNAME - the name of the function
- %RASTERX - the horizontal mesh size (\*1)
- %RASTERY - the vertical mesh size (\*1)
- %POINTS - number of points
- %AREAS - number of triangles
- %XFUNC - the function description for X axis (\*2)

---

```
%YFUNC      - the function description for Y axis (*2)
%ZFUNC      - the function description for Z axis (*2)
%OBJNAME    - the name of the objectfile (*3)
%IMGNAME    - the name of the image (*4)
%%          - an %-character

\n          - linefeed
\           - an /-character

(*1 - only for functions with an mesh
*2 - only for 3d (parametric)-functions
*3 - only for object-functions
*4 - only for image import-functions)
```

Author

Like title but this time for the author notice.

Cancel

Closes the window. The old settings are used.

Ok

Closes the window. The new settings are used.

See also: Drag & Drop

## 1.46 group\_edit3dfunction

Edit 3D-function

---



---

Belongs to the Edit-window.

Function

Formulas to calculate the X-, Y- and Z-values. They may only depend on variables u and v otherwise you get an errormessage during functioncheck.

The 'Add'-button besides the stringgadget opens a window that contains all implemented operators/functions. Simply do an doubleclick on an entry to insert it at the current cursor position.

See also Formula description for further information on formulas.

Factor

Defines the scale values for X-, Y- and Z-values. With the +/- buttons you can change the values in steps of 1.

See also Function creation for further information on scale values.

Parameter u

Settings for variable u. During calculation this variable is set to all values from 'from' to 'to'. It may also contain a Formula description (eg. '-2\*pi') but without variables.

---

Parameter v

Like parameter u but for the second variable v.

Use fixpoint-arithmetic

If selected integer mathematics is used for all calculations. This is faster on non-fpu computers. The whole transformations are much faster. Because of this the speed of the preview modes increases heavily.

Note: You can not render complex previews which such functions.

See also: Function creation

Formula description

Drag & Drop

Numerical gadgets

## 1.47 group\_editmandelfunction

Edit Mandel-function

---

Belongs to the Edit-window.

Mandelbrot

Real

Enter the minimum/maximum value for real axis.

Imaginary

Enter the minimum/maximum value for imaginary axis.

Max. iter

Maximum number of iterations per pixel.

Extra

Scale

The calculated height of the mandelbrot fractal is multiplied with this value.

Use fixpoint-arithmetic

If selected integer mathematics is used for all calculations. This is faster on non-fpu computers.

Note: You can not render complex previews which such functions.

See also: Function creation

Drag & Drop

Numerical gadgets

---

## 1.48 group\_editcsvfunction

Edit CSV-function

---

Belongs to the Edit-window.

File

Name

Name of the CSV-file.

Lines

If 'Name' contains an valid filename this textgadget shows the number of lines.

Line parsing

Defines how the textlines are parsed.

The following characters have special meanings:

%X - the number is used as x-value

%Y - the number is used as y-value

%Z - the number is used as z-value

%% - an %-character.

Spaces are ignored.

Examples

"%Y"

The whole line is converted to  
y-coordinate,  
eq: '1.00' would be a valid line.

"%X,%Y,%Z"

Every line must contain three values  
seperated by commas.  
eq: '1.0, 2.0 , 3.0'

"Data: %Y;%X;%Z"

Every lines must contain the 'Data:'  
entry and three coordinates seperated  
by semicolon.  
eq: 'Data: 1.0; 2.0;3.0'

Note: There must be at least on '%Y' within the string.

Auto mesh-sizing

If selected Plotter3D automatical calculates an mesh-size from the number of lines.

Save as reference

\*Not built-in.\*

Parameters

X

Defines special parameters for X-axis.

---

#### Data

If selected the x-values are read from the CSV-file. The 'Line parsing'-string must contain a "%X".  
If its not selected the x-values are calculated via interpolation from 'from' and 'to' values.

#### From

Defines minimum x-value.

#### To

Defines maximum x-value.

#### Factor

Defines scale-value for x-axis.

#### Y

Same like 'X' but for y-axis.

#### Z

Same like 'X' but for z-axis.

#### Use fixpoint-arithmetic

If selected integer mathematics is used for all calculations.  
This is faster on non-fpu computers.  
Note: You can not render complex previews which such functions.

See also: Drag & Drop  
Numerical gadgets

## 1.49 group\_editobjectfunction

Edit Object-function

---

Belongs to the Edit-window.

#### Name

Name of the Object-file.  
Currently only TDDD-files (Imagine objects) are supported.

#### Auto Center

If enabled the object is centered.

#### Auto Scale

If enabled the object is scaled so that is maximum width is 200.

#### Auto Naming

If enabled the functions name is got from the name of

---

the object.

See also: Drag & Drop

## 1.50 group\_editimagefunction

Edit Image-function

---

Belongs to the Edit-window.

Usual pictures have sizes from 320x200 to 1024x768 and above. Of course such resultions are hard to maintain as 3D-datas (320x200 means 64000 points and 128000 shapes!). You can use an mesh-size that is smaller than the image but the results won't be to good (use 'Blur'-option for better results). Therefore scale down the image with an paint/imageprocessing program. I recommend an size fo 100x100 pixel as I do all my final renderings with an mesh-size of 100x100 (or 200x50 etc.) points. Instead of loading an picture you can also enter an font and some text which is then rendered as an picture which is then used like an loaded picture.

Edit Image import

Input type

Determines the type of input.

Supported types are 'Picture' (loads an picture) and 'Font' (renders an picture).

Picture name

The name of the picture. The picture type must be supported by an installed datatype, a chunky-picture or a ilbm-picture.

Font

The font used to display the text.

A Fontgadget is used for selecting the font and justification.

Text

The text to be rendered. May contain format signs ('\\n'-linefeed, '\\'-backslash) enables to create multiline texts.

Color -> Height

Defines the method used for transforming the image into height steps.

Pen number

The position of the color (of an pixel) within the color lookup table is used as height. Useful for eq. greyscale images.

---

#### R+G+B

Just adds red, green and blue component of an pixel to get the height.

#### Scale

The height (0..1) is multiplied with this factor.

#### Blur

Smooth the image by the given value.  
'0' means no blurring, '8' is heavy blur.

#### Sharpen

Sharpens the contours of an image. Simply calculates  $\text{height}^{\text{sharpen\_value}}$  (power).

#### Keep aspect

Scales the mesh to the same aspect as those of the image.

#### Picture information

Opens an requester showing width, height and number of colors of the picture. There you can also set the mesh-size according to the image resolution.

#### Use fixpoint-arithmetic

If selected integer mathematics is used for all calculations. This is faster on non-fpu computers.

Note: You can not render complex previews which such functions.

#### See also: Drag & Drop

Numerical gadgets

## 1.51 node\_functioncreating

### Function creating

---

#### Notes about function creation:

Plotter3D calculates for every function the minimal and maximal X-, Y- and Z-coordinates. The internal scale value is calculated in the following way:

- choose the axis with the largest extension (maximum-minimum)
- the scale value is 200 divided by this extension

This simply means that every function has a maximum extension of 200 units.

Because this is done *after* calculating you won't get more details. If you use fixpoint-arithmetic you may also get unusable results.

The way is to set the scale values for every axis to an appropriate value because they are used *during* calculating.

---

Example 1:

Parameter U: from -pi to pi  
 Parameter V: from -pi to pi  
 Function-X : u  
 Function-Y :  $\sin(u^2) \cdot \cos(v^2)$   
 Function-Z : v

The X- and Z-values should have an ideal extension from -100 to 100 ( $100 - (-100) = 200$ ). Therefore set X- and Z-factor to 23 ( $100/\pi$ ).  
 The Y-values reach from -1 to 1. To see anything useful you should set the Y-factor to 20 (or similar).

Example 2:

Parameter U: from -pi to pi  
 Parameter V: from 0.0 to pi  
 Function-X :  $\sin(u) \cdot \cos(v)$   
 Function-Y :  $\cos(u)$   
 Function-Z :  $\sin(u) \cdot \sin(v)$

If you take a closer look you see that the X-, Y- and Z-values reach from -1 to 1. Set all factors to 100 and you get an perfect sphere.

## 1.52 node\_formula

Allowed formulas

---

A formula consists of any number of numbers, variables, constants, operators, functions and brackets.

Numbers : eq. 2, 1.0, -3.1459  
 Variables : u,v (case doesn't matter)  
 Constants : e,pi,c,g (case doesn't matter)  
 Brackets : (, )  
 Operators : (order like their priorities)  
     ^     x to the yth power  
     \*     multiplication  
     /     division  
     %     remainder of integer division  
     \     integer division  
     +     addition  
     -     subtraction  
 Functions : only functions with one argument  
     the argument must be surrounded by brackets (eq.  $\sin(u^3)$ )  
     for implmented functions see 'Add'-buttons of the edit-window.

correct formulas are (for example):

u  
 -pi  
 $u - u^3/3 + u \cdot v^2$   
 $(15 + 6 \cdot \cos(u)) \cdot \sin(v)$

---

$$(1 + \cos(20 * (u^2 + v^2)^{0.5})) / (13 * (u^2 + v^2) + 0.5)$$

## 1.53 node\_previewtypes

Previewtypes

---

Plotter3D has the following previewtypes:

Hiddenline

Draws the borders of all areas. Hidden parts are overwritten.  
With this mode you may get errors if big/long areas are drawn.

Random color

All areas are filled with a random color.

XYZ-colorflow

Every point of the mesh gets a special color. The color is calculated from the coordinates of this point. The red share is proportional to the value of the X-coordinate. For the green respectively blue share the Y respectively Z-coordinate is used. Note: the original coordinates are used which means that the colors are also rotated, scaled, moved. While drawing the colors are spread which results in a smooth colorflow.

Heightsteps

Every point the mesh gets also a special color that is spread on the area.  
The color depends only on the Y-coordinate. It results in a rainbow-effect. Note: again the original coordinates are used.

Flatshading

This mode needs a lighting.  
For every area a color that depends on lighting and material is calculated. Then the area is filled with this color.

Gouraudshading

The colorvalues are calculated for every point of the mesh and then spread on the areas.

Phongshading

\*\*\* Warning, this mode needs much calculation time \*\*\*  
The special thing with Phongshading is that the lighting is calculated for every pictures \*during\* rendering. This slows down calculating speed very much but increases quality of the pictures.  
Note:  
Because its not possible to put the colorvalues in an valid range, larger values are truncated.  
Of course this may result in wrong colors.

Colortest

---

This is a testmode for the colorsystem. It creates a colorflow from blue, red, yellow and green.

See also: Complexpreview

## 1.54 node\_boundingBox

BoundingBox

---

Plotter3D calculates for every function the minimal/maximal X-, Y-, Z-values. The boundingbox is then a cube with these corners. Hidden edges are drawn ghosted. Also origin and axis are marked.

## 1.55 node\_lightsources

Lightsources

---

Plotter3D supports the following lightsources:

### Ambient light

You should only define one ambient lightsource. Ambient light is the diffus reflected part of light and lights *\*all\** areas with the same intensity. You can change the general brightness and color of the picture.

### Parallel light

This light is like sunlight. Because of the large distance the rays are nearly parallel. Depending on the angle of incidence the intensity spreads on a more or less large area. This also causes highlights.

The preview of this lightsource is an line, showing the direction of the light, and the name of the lightsource.

### Point light

This lightsource emits lights into every direction. The intensity decreases with the distance. At a special distance the light is so faint that it can be ignored. You can enter this maximum lighting distance as radius of the lightsource.

The preview of this lightsource is an point with connections to a circle (the light radius). Besides the name is displayed near the circle.

### Spot light

\* not implemented yet \*

---

See also: Settings/Preview

## 1.56 node\_colorcontrol

Colorselection

---

Plotter3D calculates most datas in 24Bit (16.8 million colors). The most graphic cards features special 15/16/24Bit-modes which are supported by CyberGfx. On such a screen Plotter3D can display the datas directly. Normally there are less free colors. To display the pictures the number of colors must be reduced. The main goal was high speed because Plotter3D displays only a preview. To get high quality results save the 24Bit data and use specialized programs like SuperView or AdPro for color reducing.

Plotter3D has 2 different modes:

Grayscale-Matrix

In this mode the datas are first merged to 8Bit (256 greyscales) and then reduced to the number of free colors.

RGB-Matrix

Now the colors are reduced directly to the available color number. This mode needs at least 32 free colors otherwise it switches back to grayscale mode.

Because of the internal construction Plotter3D can only use a fixed palette. Means even if the picture has only blue tones the palette also contains red and green colors.

Internal

The color management and the representation is independent from rendering. Plotter3D has 3 specialized "drivers" for this. The currently used driver and his datas are shown in the System-window (Menu Project/About...).

CyberGfx

This driver is used with 15/16/24Bit Cybergfx-screens. Because the 24Bit-datas are given directly to CyberGfx-system it is very fast.

Kick v39+

This driver is used if you have installed Kickstart 3.0 (or above) and use a screen with <=256 colors. The driver tries to obtain as much free colors as possible. Not used pens are freed. It has the capability to render into HAM6 or HAM8 screens.

Kick v36+

This driver is selected if no other driver can be used. Because its not possible to obtain colors, no preview can be drawn on a public screen. On custom screens (like the one

---

used with Rendering) all colors are used.  
It has the capability to render into HAM6 screens.

See also: Settings/Buffer

## 1.57 node\_savefiles

SaveFiles

---

Plotter3D uses the IFF-fileformat for all files (except picturetypes like Targa). This has the advantage that a file must not contain all information or may contain additional ones. If some information are missed they are replaced by internal defaults. For example a function-file may have no lightsources. On the other hand its possible to load a function-file as material definition. Plotter3D simply searches the material datas within the file and use them.

The program 'Analyse\_Savefile' (in Plotter3D:Tools/) shows all information ("Chunks") of file that was created by Plotter3D. This makes it easy to check what Plotter3D writes into a file.

## 1.58 node\_dragdrop

Drag&Drop

---

MUI has a standard way of handling drag & drop: If you grab (press LMB and hold it) an draggable object MUI creates an image of that object which you can drag around. If you move to image over an object that accepts the drop an checked rectangle is drawn around it. Just release LMB and the drop is performed.

The D&D Clipboard is the heart of Plotter3D's D&D facilities.  
It can hold any draggable objects.

As may have already noticed not every object can be dropped on another object. This is logical because its senseless to drop an color definition on an string object.

Currently Plotter3D defines the following D&D types:

Name	Originator
Color	Color gadget. Just grab the popup button that contains the colordisplay. D&D Clipboard displays the red, green and blue value.
Number	Numberstring gadget. You must hit the (small) boarder of the string gadget to start D&D. This due to an

MUI-Limitation which may be removed in future versions of MUI.  
D&D Clipboard displays the value.

Function Function listview of Control window. Simply grab an entry of the list.  
D&D Clipboard displays the function name.

Lightsource Lightsource listview of Edit Lightsources window. Grab the entries of the list.  
D&D Clipboard displays the lightsource name.

Material Material window. Just start D&D by pressing LMB over the DragDrop-Button at the top of the window.  
D&D Clipboard displays the function name to which the material belongs.

Path Pathstring gadget. Again hit the boarder to start D&D.  
D&D Clipboard displays the path.

String General string gadget. Same method as with all string gadgets.  
D&D Clipboard displays the string.

Transformation Transformation window. Just start D&D by pressing LMB over the Drag&Drop-Button at the bottom right edge of the window.  
D&D Clipboard displays the function name to which the transformation belongs.

## 1.59 node\_problems

### Problems

---

P: Plotter3D crashes if you press the HELP-key.

S: I had an version of amigaguide.library that crashes immeditly if you wanna open an database on a screen with more than 5 bitplanes.  
Some programs told that it was version 39.11 but 'c:version' said it is version 34.6.  
Replace it with an newer one and everything works fine.

P: Plotter3D does not draw an complex preview. Only a black area is shown.

S: This may have different reasons.

On Kick 2.x you will never see an complex preview. This is because of the lack of some essential system routines.

If you running Kick 3.x you should have a look at the About window. It shows how many free pens could be obtained. If this value is zero the used PublicScreen has not enough free pens. Simply open the MUI-Preference window (menu 'Settings/MUI...'), go to the 'System'-page and select a new PublicScreen. An good idea is to define a new screen with MUI's PSI which is only used by Plotter3D.

---

## 1.60 gadget\_color

Color gadget

---

This gadget is used to define a color. It consists of an text gadget that displays the red, green and blue value (0..1) and an button with an colordisplay. If the screen has free pens the colordisplay shows the actual color otherwise the best matching color is shown. Kick 2.x user will see only a hatched rectangle.

If you press the button a page pops up where you can change the color. Press the button again to close the popup.  
Depending on the screen you may get different versions of this page:

Standard version

The page contains just three sliders where you can change the red, green and blue values in range from 0 to 255.

Colorwheel version

The page contains sliders for red, green and blue values in range from 0 to 255. Besides it has an colorwheel, an gradient slider and an popup button where you can set predefined colors.

This version normally pop up only on High/True color CyberGfx screens. The reason for this behavior is that the colorwheel obtains a lot of pens for his display. Because Plotter3D can open many windows which may contain color gadgets, only the first one will look good. The others then just display a black wheel with some letters.

You can force Plotter3D to use colorwheels by setting Global preferences/Always use colorwheel.

See also: Drag & Drop

## 1.61 gadget\_numerical

Numerical gadget

---

An numerical gadget has an decrease and a increase button beside it. Press the left button to decrease the value of the gadget or the right one to increase it.

You can change the button images with Imagebutton Preference/Buttons.

An numerical gadget may be anything; even an slider. But normally it is an floatstring (displays an float value) or integerstring gadget (displays an integer value). Both gadget have the same features:

Own edithook

This adds some extra features to the stringgadget. Press

---

Cursor up to decrease the value of gadget or Cursor down to increase it. This is the standard behaviour of all numerical gadgets and does the same like pressing the decrease/increase buttons.

#### Error check

Whenever you leave an stringgadget it checks the contents. If it contains an illegal input (eq. '1.0.') an bubblehelp appears. As soon as you press any key it disappears.

#### Range check

Whenever you leave an stringgadget it looks if the value lays within the valid range. If not then a bubblehelp appears showing you the valid range.

See also: Drag & Drop

## 1.62 gadget\_font

Font gadget

---

An font gadget consists of an text gadget, displaying fontname and fontsize, and an 'Select...'-button that opens the popup. The popup-window contains all settings of an font gadget: (Note: unnecessary gadgets are invisible)

#### Enabled

Enables/Disables the rendering.

#### Font

Defines the used font. Click on the popup-button to open an standard system fontrequester where you can select an font. All fonttypes are supported including colorfonts. Depending on the fonttype some settings may be disabled.

#### Text

The color of the Text of this font.  
A color gadget is used for this.

#### Backgr.

The background color.

#### Mode

Defines how the text is rendered:

##### Text

Only the text is drawn. The background is not modified.

##### Text+Background

Draws the text and fills the background with the background color.

---

Justify X

Defines how the horizontal justification of the text.

Justify Y

Defines how the vertical justification of the text.

Close

Closes the popup. Same as pressing the 'Select...'-button again.

---