

What's new in CPFG

Radek Karwowski

Contents

I	ver. 4.0	3
1	New L-system elements	3
2	New features in the view file	3
3	Textures with alpha channel information	3
4	Function specification file	3
II	ver. 6.0 (L-studio 2.0 beta)	4
III	ver. 6.1 (L-studio 2.1)	4
5	New UI elements	4
6	Continuous modeling support	4
7	New rendering features	5
8	Curves	5
9	Function galleries	5
10	Other changes	5

IV L-studio 2.1(.2)	6
V ver. 6.2 (L-studio 3.1)	8
11 New UI elements	8
12 New view file commands	8

Part I

ver. 4.0

1 New L-system elements

Function `func(id, x)` takes two arguments. The first argument is the identifier of the function (see description of the `function` command in the next section). The second argument is a number $x \in [0, 1]$ which is the argument of the function. `func` returns the value y of the function.

2 New features in the view file

`function: file [n]` defines a function specified by a function file. By convention function file names are of form `filename.func`.

- if the optional parameter n is specified then `cpfg` will precompute n values of the function, evenly spaced along X . Whenever the function `func` is used the returned value is a linear interpolation between the two closest precomputed values.
- if the parameter n is omitted then the values of the function will be calculated for each call of the function `func`.

The argument *id* (the first argument of the function `func`) is one-based ordinal number of the `function` label in the view file.

3 Textures with alpha channel information

`cpfg` supports textures with the alpha channel information if the texture is stored in SGI RGB(A) file format.

4 Function specification file

A function specification file defines a function as a spline curve which the first point X coordinate is equal to 0 and the last point X coordinate is equal to 1. Also for any two points p_i and $p_{i+1} \Rightarrow X(p_i) \leq X(p_{i+1})$

Function specification files usually have extension `func`. The format of the files is:

```
range:  0.0 1.0
points:  n
x1 y1
```

$$\begin{array}{cc} x_2 & y_2 \\ \cdot & \cdot \\ x_{n-1} & y_{n-1} \\ x_n & y_n \end{array}$$

Part II

ver. 6.0 (L-studio 2.0 beta)

- **Independent scaling of surfaces.** $\sim l[(sx[, sy, sz])]$. If no parameters are present then the surface is not scaled. If only one parameter is present, then the surface is scaled uniformly by the factor sx . If three parameters are present the surface is scaled by sx along X, sy along Y and sz along the Z axis.
- **BMP textures support.** Textures can be 24-bit .bmp files
- **X module inserted in the string.** When the user clicks with the Ctrl and Shift buttons pressed `cpfg` will insert module X before the module that draws the primitive selected by the user.

Part III

ver. 6.1 (L-studio 2.1)

5 New UI elements

- **View panning in parallel projection** Left mouse click + Shift pressed start panning.

6 Continuous modeling support

The following commands can be called by external programs by sending WMCOMMAND message to the `cpfg` window.

- Reread colors
- Reread contours
- Reread surfaces
- Rerun
- Reread functions and rerun

- Reread curves and rerun

In addition two predefined functions have been added to L-system:

- `GetDerivationLength(dummy)` and
- `SetDerivationLength(steps)`.

The first one returns the number of derivation steps in the current L-system. The second one sets the number of derivation steps in the current L-system.

7 New rendering features

- **Antialiasing mode.** A new entry to the view file has been added. `antialiasing: on|off`. Currently antialiasing is working only for lines.

8 Curves

(Lars)

9 Function galleries

A new way of using function objects has been introduced in addition to the one found in the `cpfg` version 4.0. Instead of storing each function in a separate `.func` file the user might specified a *function gallery* file. The corresponding entry in the view file is: `function set: filename`.

If `cpfg` finds this entry then all functions from the gallery file are read. The function id's are 1-based and correspond to the order of the functions as they appear in the gallery file. In addition when preprocessing the L-system file `cpfg` adds a number of `-DFUNC=id` commands to the preprocessor. The *FUNC* identifier is the capitalized version of the function name as it appears in the gallery file. *id*'s are id's of the corresponding functions.

Number of precalculated samples is now stored in the gallery file together with each function. For full specification of the function fallery format see *Cpfg, L-studio file formats*.

In addition, function format v. 1.01 (see *Cpfg, L-studio file formats*) is also supported.

10 Other changes

The following commands can be called by external programs by sending WMCOM-MAND message:

- New render
- New projection

New render rereads the color files (colormap or material file), view file as well as all files referenced to in the view file (surafces, contours etc.) but no view (projection) parameters are modified. It then redraws the scene without changing any projection parameters.

New projection rereads the color files, view file and files referenced to in the view file (surfaces, contours, etc) but no drawing parameters are mofigied. It then redraws the scene after having recalculated new projection parameters.

Also some bugs in the parser have been fixed.

Part IV

L-studio 2.1(.2)

- **Find in text editors.** It is possible to search for a string in all the text editors (L-system, View, Description, Text file). To open the *Find* dialog press `Ctrl+F`.
- **Drop shortcuts** Shortcuts to directories can be dropped on the main L-studio window.
- **Continuous modeling mode.** Command `Continuous modeling` under the `Options` menu turns on and off the continuous modeling mode.

Continuous modeing mode affects the following editors:

- Colormap editor
- Material editor
- Surface editor
- Curves (advanced surface) editor
- Contour editor
- Function editor
- Panels (in the execute mode)
- **Reread text file** If you want to quickly reread the file that is currently opened in the `Text file` editor double-click on the filename (to the left of the current line number)
- **Colorschemes for some editors.** It is now possible to specify colors for some editors. Depending on the current editor active the command `Customize` from the `Options` menu will open a dialog box that makes it possible to modify the appearance of the current editor.

In case of text editors the user may change the font and the size as well as the background and text colors.

In case of the Contour and the Function editors the user can change the colors of the following elements:

- Background
- Grid
- Axis
- Curve
- Line segments
- Control points
- Labels

The same settings apply also to the corresponding galleries. The customization of the surface editor is limited and not quite consistent.

- **Tabs in the project window** might include labels and/or icons. To change this setting choose the *Settings* command from the *Options* menu.
- **Function flip and galleries.** A new format for storing the functions has been introduced (see *Function galleries*). It makes it possible to flip the functions plots, so that they look like $f(y)=x$. This information is stored in the function file if it is stored in the function format v. 1.01. This format is required if the functions are to be stored in a single gallery file.
- **Tooltips in the galleries.** When pointing on an element in a gallery a tooltip will show up and display the name of the object. This applies to surfaces, contours and functions.
- **Smarter scrolling of value sliders** in the *Simple surface editor*. If the user clicks on one of the value sliders in the surface editor and drags the pointer outside the window (to the left or to the right) the slider starts scrolling. The speed depends on how far the user has moved the mouse.
- **Drag and drop of elements in the galleries.** Elements might be moved within a gallery using *drag-and-drop*. This applies to the following galleries:
 - Material
 - Surface (simple and advanced)
 - Contours
 - Functions

- **Smart wait** in the continuous modeling mode. If it takes more than 0.1 second for `cpfg` to regenerate the image L-studio will not attempt to send further updates to `cpfg`, unless the modification made by the user can be considered as "final" (e.g. the user has released the mouse button after dragging a control point). L-studio will wait infinitely for `cpfg` to complete redrawing, but if the redrawing takes more than 1 second then the user is presented a dialog box that makes it possible to abort waiting for `cpfg` to finish redrawing.
- **Smart apply**. Whenever a change in an editor is transferred to the gallery (by selecting the `Apply` command, or by selecting another element from the gallery) `cpfg` is notified about the change, so that there is no need to explicitly select the `New View|Render` command.
- **New projection and New render** command.
- **Experimental Pick color dialog** in the material editor. L-studio is now using a new Pick color dialog box, based on the HS(V) circle together with the sliders working in the RGB and the HSV spaces. Pick color dialog is modeless.

Part V

ver. 6.2 (L-studio 3.1)

11 New UI elements

- **Select output directory** menu item makes it possible to specify a directory where all output files will be created.
- **Keyboard shortcuts** are available for the commands that control the animation:
 - Ctrl+F - Step
 - Ctrl+R - Run
 - Ctrl+V - Forever
 - Ctrl+S - Stop
 - Ctrl+W - Rewind

12 New view file commands

- **zoom min** and **zoom max**.
`zoom min: zmin`
`zoom max: zmax`

These commands make it possible to control by how much the user can zoom in or out the view these commands (if present) take precedence over the default values which are 0.05 and 50 respectively.