

release qmga_r374

1 Installation

Right now there is no easy, out of the box solution to install QMGA, sorry.

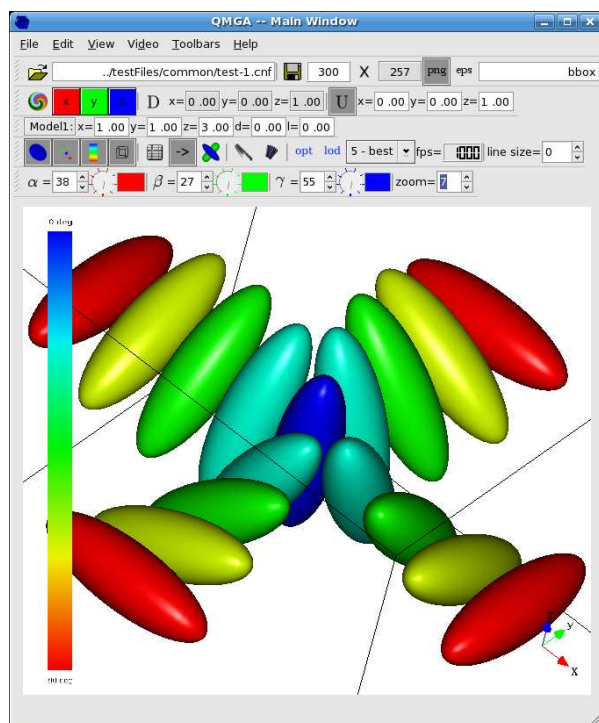
See the README file for some hints. If it this does not help you, feel free to contact me using the following email address:

qmga@users.sourceforge.net

2 Usage

2.1 General

Here we see a screenshot of QMGA.



2.2 A Word on File Formats

The number of implemented file formats in QMGA is ever growing, because many people use their own written programs to do their simulations. In the following two examples are described.

gbmega

The format looks like follows:

- – header:
- int (number of molecules)
- double (x side length of the unit box) (or three doubles)
- double (y side length of the unit box) (or three doubles)
- double (z side length of the unit box) (or three doubles)
- 2 doubles (for moving boundary conditions)
- – molecule information:

- 12 doubles (ri,vi,êi,ui), int (label), int (type tag, optional) where ri is the position of molecule i, vi is its velocity, êi is its orientation, and ui is its orientation velocity.

Of these numbers, only the box sides, ri, and êi, are used for visualization purposes. The program can deal also with noncubic unit boxes; in this case, a 3×3 matrix must be input to specify it. As an example look into “mga_dummy.cnf”, “mga_dummy-mult.cnf” and “mga_dummy-nocubic.cnf” which are located in the QMGA main directory.

lammps

- ITEM: TIMESTEP
- int
- ITEM: NUMBER OF ATOMS
- int
- ITEM: BOX BOUNDS
- 2 doubles
- 2 doubles
- 2 doubles
- ITEM: ATOMS
- int (number), string (type), 7 doubles (ri, êi) where ri is the position of molecule i and êi is its orientation as a quaternion.

See the file “mga_dummy-lammps.cnf” for an example. Notice though, that one has to cut a trajectory into different files, one for each timestep. Meaning, right now QMGA can only handle one single configuration in each file. This might change in the future.

Notice also, that the output format of lammps is user defined. QMGA adopted the format as introduced by Mike Brown (Sandia) in the lammps example files for the GayBerne ellipsoids. (tag type x y z quatw quati quatj quatk)

Other Examples can be found in the QMGA install folder (mga_dummy-*.cnf).

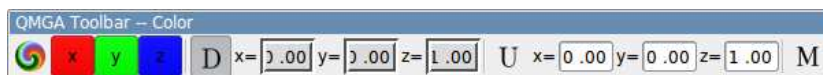
2.3 Command Line Options

QMGA knows some command line switches, which are to be described in this paragraph. All of them are optional since most have default values.

- -f

- format of the given konfiguration file these include at this time:
 - gbmega
 - lammmps1
 - lammmps2
 - gbmegaBiax
 - cinacchi
 - The format for the next opened file can also be set via the GUI.
- -i
 - input file name
 - if none is given the last opened file is used
 - if none was opened yet, a dummie file is loaded
- -c
 - optional (defaults to “color-090.map”)
 - color map filename
- -m
 - optional
 - models file name
 - a models file can also be loaded and saved via the GUI
- -v
 - video settings
 - must be followed by a filename and 4 integers (startvalue stopvalue stepvalue and #digitsvalue, see 2.7)

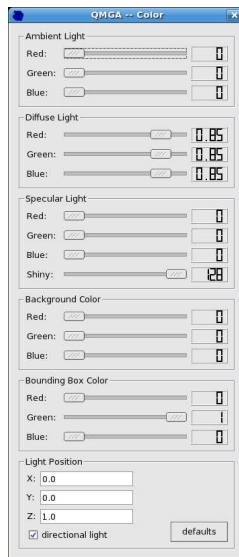
2.4 Toolbar: Color



2.4.1 OpenGL Color Settings

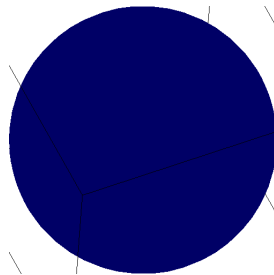


This button opens the following dialog:

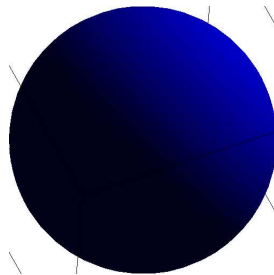


Here it is possible to make settings on the color scheme used by OpenGL.

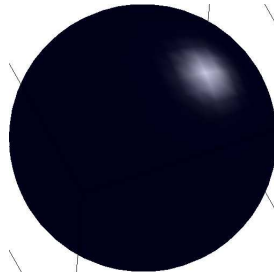
The following picture shows the effect of ambient light on a sphere. It comes from all directions and is reflected in all directions, thus destroying any 3D impression:



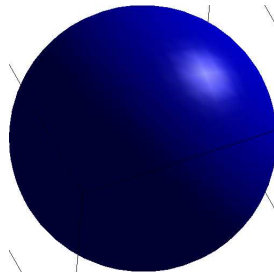
The next picture shows the effect of diffuse light, which comes from a certain point in space and is reflected in all directions.



Again the next picture shows the effect of specular light, which comes from a certain point in space like diffuse light, but is reflected like on a mirror.



If all light sources are put together and the intensities are somewhat balanced the sphere looks like this.



The value for “shininess” decides, how big the reflection appears.

2.4.2 Color of the Coordinate System



Each of these three buttons opens a dialog to pick a color. The chosen color is then applied to the according axis of the small coordinate system inside the render area as well as to the button itself and some other elements linked to the corresponding direction.

2.4.3 Color Axis



The first set of coordinates shows the director of the mesophase as calculated beforehand. The second can be edited by the user. The two buttons “D” and “U” define, which vector is used for colorization purposes. The “M” Button sets the colorization thus, that it is determined by model as set in the models parameters overview.

2.5 Toolbar: Camera Position



This toolbar contains four sets of controls. Three for angle manipulation and one for zoom.

2.5.1 Angle Setting



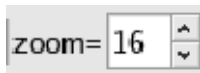
Here α represents the rotation around the x-axis in a chain of three successive rotations (α : x-axis, β : y-axis, γ : z-axis). The following description is the same for all three angles.

The spinbox widgets are continually updated when the objects are rotated. The boxes can also be used to spin the objects, but as they represent rotations executed in a chain, the actual axis around which the individual rotation takes place is NOT the axis one might guess on first thought. The reason for these spinboxes is to recreate a certain angle triplet (α, β, γ) . Until now there is no built-in functionality to save such triplets other than the one used at the time of closing QMGA which is restored on the next start. Others triplets have currently to be saved the old fashioned way (pen and paper).

The dial can be used to rotate the objects around the x,y,z axis as shown by the small coordinate system in the render area (if activated).

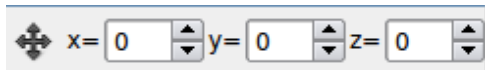
The purpose of the line edits is to rotate around the corresponding axis by a certain angle. To do so one writes the desired rotation angle into the line and presses “enter”. It is possible to enter negative angles.

2.5.2 Zoom Setting



This spin box shows and sets the actual zoom value. The value is saved/restored with program termination/startup. The value is also continuously updated, when the zoom is changed e.g. by mouse wheel usage.

2.5.3 Translation

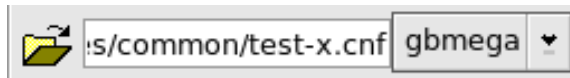


If the arrows button is activated the view of the system can be translated along the coordinate axes. This is useful if a part is to be viewed at a higher zoom, so that the interesting part is outside of the window. Using this feature it can be translated back into the viewer.

2.6 Toolbar: Open/Save



2.6.1 Open Configuration File



The button opens a system file browser, but the path can also be typed directly into the line edit.

Notice: There is no restriction as to how the loaded file has to be called (e.g.: no specific .xyz extension is required), so be sure that the file is the one you want to load. Wrong file formats currently result in an error message on the command line.

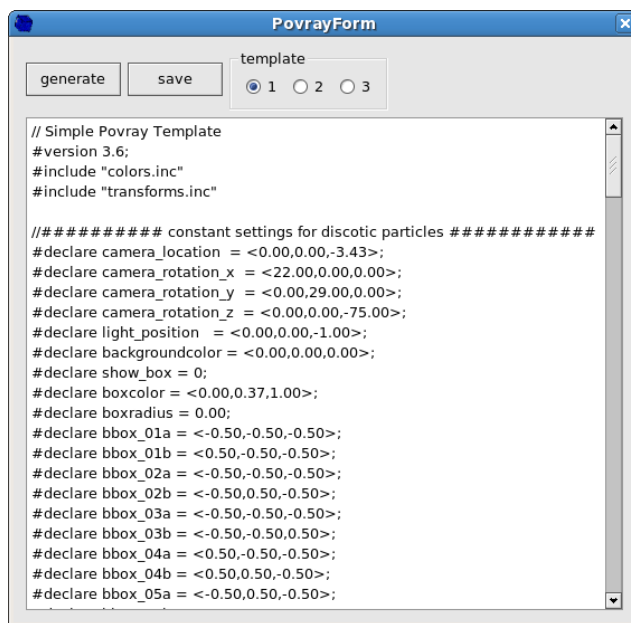
The drop down list which shows “gbmega” on the picture is used for selecting the file format (meaning the parser function) to use when opening the next file.

2.6.2 Save Povray File



– NOTE: Due to major internal changes of QMGA the povray feature is not available in the actual version. This will hopefully be remedied in a future release. –

This button opens a new window:

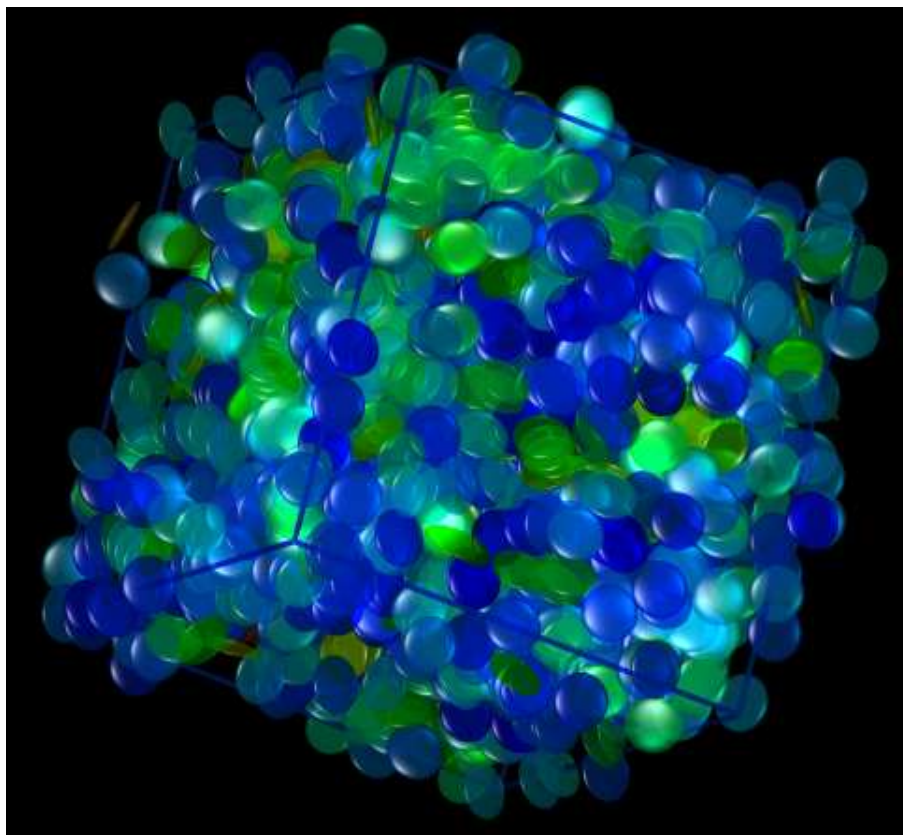


Pushing the “generate” button generates output similar to the picture above. All output is editable if one knows how to handle povray scene files. By now there are three preset povray files in the qmga folder, which can be selected by

the radio buttons “1”, “2” and “3”. These files “povray.template.0x” can of course also be adjusted.

Sadly the rotation angles shown in QMGA seem not to be compatible with povray. I hope that I can solve this issue in the future. As it is one has to adjust the values for “#declare camera_rotation_x,y,z” manually if a certain viewpoint is desired.

Here is an example of preset 2 which results in a rather glassy output (whatever that is good for... but it looks nice.):

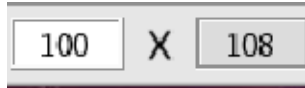


2.6.3 Save Screenshot



This button opens a small message box giving the name of the file(s) to be saved and asks permission to do so.

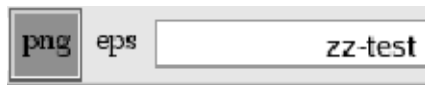
2.6.4 Size of Screenshot



These two line edits are used to define the size of the screen shot (in pixel units). One of the line edits is grayed out were the other can be written into. The value of the grayed line edit is calculated using the aspect ratio of the render area. While resizing the program window the editable value stays constant and the grayed is constantly recalculated as the aspect ratio changes.

The “x” symbol is actually a button that toggles which of the line edits is editable and which is not.

2.6.5 Name and File Format of Screenshot(s)



If the toggle buttons “png” and “eps” are turned on (either one of them or both) a file of the appropriate format is saved to disk, when the save button is pressed. An error box appears if no file format is specified.

The line edit holds the base name that is to be used for the saved file. It is extended by .png or .eps depending on the format.

(Note: Two files are saved if both formats are specified, of course...)

2.7 Toolbar: Video Settings

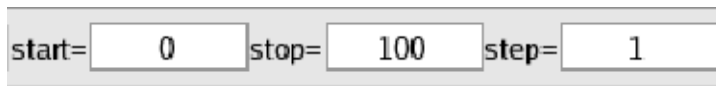


2.7.1 Opening Video File



This button and line edit work in the same way as the “normal” open file. The only difference is the handling of the specified file.

2.7.2 Video File Range



The “player” assumes, that all successive files are named the same except for an .xxxxxxxxxxx index number at the end.

The three line edits shown here specify with which file to start, with which to stop and how to increment the index.

Example:

Loaded ../test/dummy.0000

The whole directory contains files up to dummy.5000.

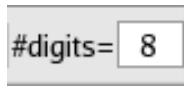
start = 0; stop = 1000; step = 1

This results in the playback of all files in the range from dummy.0000 to dummy.1000.

If, for example, step is changed to 10 only every 10th file is loaded (0,9,19,...).

This functionality is useful to avoid error messages when the index is not continuous and to speed up replay, when the timescale is too large.

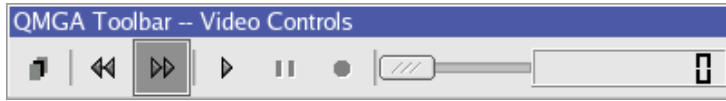
2.7.3 Number of Digits



This line edit specifies the index endings number of digits. The value is internally used and quite important.

e.g.: If the file is called file.00000000 #digits is 8.

2.8 Toolbar: Video Controls



2.8.1 Capture



If this toggle button is on, all played frames are written to disk with a name (and position) based on the one specified in the screenshot line edit.

2.8.2 Direction



These buttons decide whether the trajectory is played forward or backward.

2.8.3 Standard Controls



Play, Pause, Stop. Not much to say about these.

2.8.4 Progress-Slider and LCD



The slider shows the overall position of the playback based on the input made in “start” and “stop”. It can also be used to seek to another position in the trajectory e.g. by dragging it with the mouse. The LCD shows the number of the file that is currently shown.

2.9 Toolbar: View



2.9.1 Toggle Full Render, Coordinates and Colormap



The first toggle button switches between full and stick render view. The second button switches the coordinate system in the lower right corner on and off. The third button switches the colormap on and off and the last button switches the bounding box representation on and off.

2.9.2 Model Parameters and Switch Models



The leftmost button showing the folder icon opens a file dialog. Here a file with model specifications can be chosen to load.

The second button from the left with the disk icon also opens a file dialog. This time a file(name) can be set to save the current model setup to. This is especially useful if systems with many different models are used and/or if the model setup changes very frequently. In those cases it would otherwise become quite tedious to enter all the parameters again and again...

The third button showing the list icon opens the following dialog:

QMGA -- Models Parameters																	
	model	x	y	z	d	l	rad sphere	rad circle	rad sphere	cut	rad sphere	angle	wire	force	red	green	blue
1	ellipsoid	1.00	1.00	0.20	1.00	2.00	0.50	0.50	0.10	0.50	1.14				255	50	50
2	spherocylinder	0.00	1.00	0.00	3.00	1.00	0.50	0.50	0.10	0.50	3.14				50	50	255
3	spheroplatelet	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
4	cut sphere	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
5	eyelens	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	1.00				150	150	150
6	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
7	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
8	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
9	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
10	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
11	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
12	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
13	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
14	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
15	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
16	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150
17	ellipsoid	1.00	1.00	1.00	0.50	1.00	0.50	0.50	0.25	0.50	3.00				150	150	150

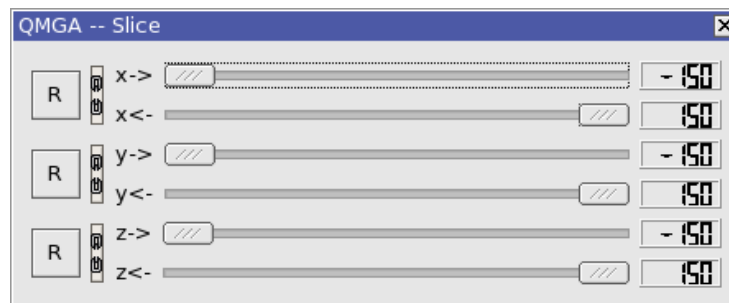
In this table the user is able to define all parameters for all models that are in use. The first two lines are in sync with the two toolbars on the main window and are always shown here. All additional lines depend on the systems shown so far, because if a loaded system needs more models as are defined right now new models are created. On the other hand if a new system needs fewer models NON are removed! All models after the first two are initialized as spheres with radius 1 and are not automatically saved to file at program termination. Note again, that when a new file is loaded that makes use of fewer models than the file before the unused models are shown here nonetheless. No model is deleted while the program is running, but if a file is loaded that needs more models new ones are created and old ones are overwritten.

The fourth button with the arrow icon enables/disables the last button which is a switch to toggle the whole system between model 1 and model 2. This switch overwrites the model settings made in the configuration file. The whole system is imposed with either model 1 or model 2.

2.9.3 Slice and Fold



The left button (knife) causes the render area to react to the settings in the slice window:



Every slider has a range of thrice the bounding box to be able to also slice systems that are shown in an unfolded state. The leftmost button in each row

of the dialog resets the slider settings and the second one fixes the distance between two sliders of the same direction.

The right button (accordion) folds an unfolded system into the bounding box, when enabled and unfolds it again, when disabled. Note that it is not possible to unfold a system that is saved in a folded state.

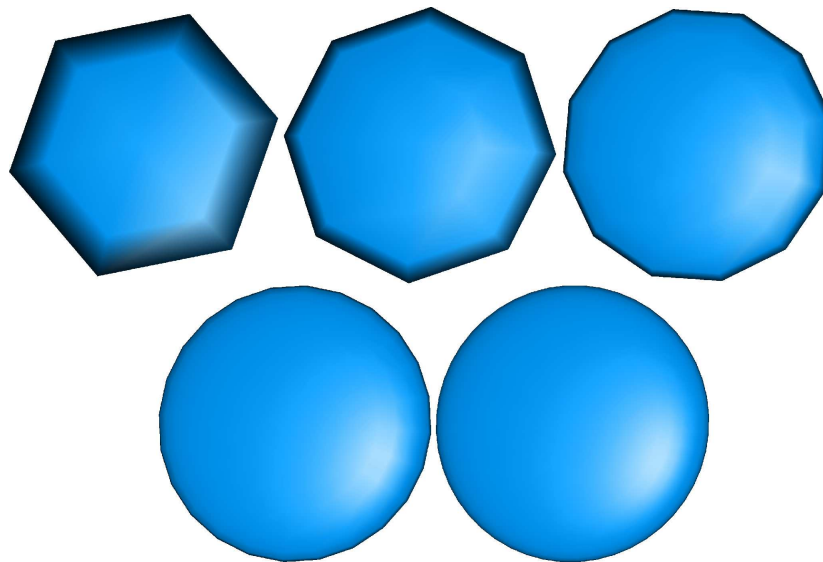
To do so anyway a trajectory has to be given where the first file has to start entirely in the bounding box. This functionality is not yet implemented, but may follow in a later release.

2.9.4 Optimization



The first button (opt) enables some optimization routines which help to speed up the visualization of large closely packed systems (starting about 10^4 Particles maybe less). With smaller systems visualization will become slower (!). So it is best to try out, if necessary (e.g. when rendering is slow...). The second button (lod) enables automatic level of detail adjustment, when the render performance is poor. Be aware, that this is not an immediate effect. It takes some render cycles to gradually lower the render quality starting from back to front. The value of “box grid” is the number of subboxes the system is divided for the optimization routine. Changes here should be made very cautiously and only if necessary.

The toolbox sets how detailed each object is represented. The range is from poor to good in five steps. In a normal session (whatever normal is) it is sufficient to work with a medium setting of 3. On a printout it might be desirable to use 5 for smooth edges. Poor is only useful for very large systems or for very slow hardware. The next picture shows the difference:

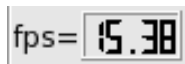


One might want test the settings by checking the frames per second toolbar.

2.10 Toolbar: Frames per Second (fps)

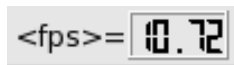


2.10.1 Current fps



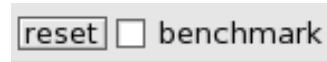
This display shows the current frames per second as calculated by the render engine. The accuracy for very high frame rates is not very good because of the limitation to one millisecond time quantization in the measurement. The highest value shown is thereby 1000 fps, where the measurement is capped.

2.10.2 Mean fps



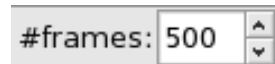
With every new fps value the mean is taken over all values, to increase the accuracy if one wants to know how fast a system is rendered.

2.10.3 Reset and Benchmark



The reset button resets the mean fps value to zero. If the benchmark checkbox is checked it also starts a series of random rotations which might be used for benchmarking purposes.

2.10.4 Number of Frames



The value of this spinbox gives the number of frames shown (rotations done) when a benchmark is started.

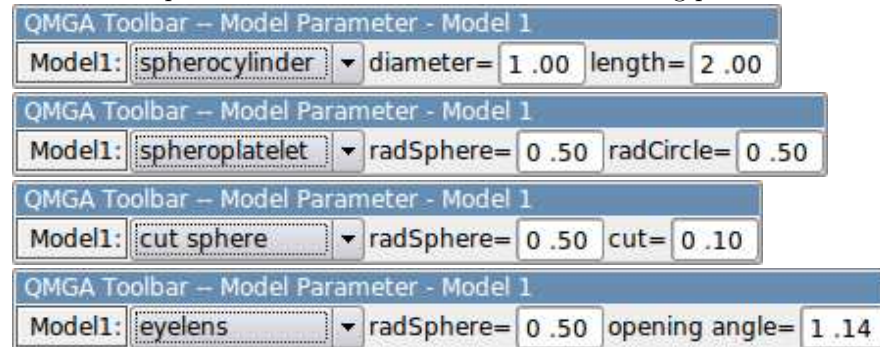
2.11 Model Parameter



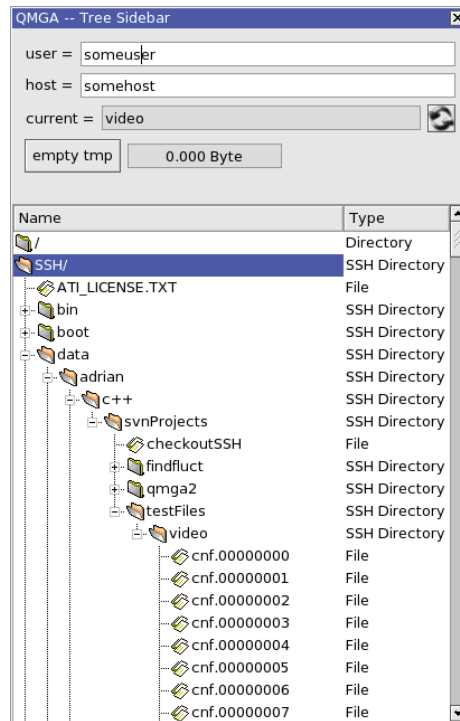
x,y,z: These are the parameters used for ellipsoids. E.g. $x=y=z=1$ results in a sphere with a diameter of 1. $x=y=1; z=0.2$ results in an ellipsoid made of a sphere by flattening it along the z-axis by a factor of five.

Other models of course have other parameters to describe them. The toolbar will change and show the correct parameters if another model is selected via the dropdown menu.

The other implemented models are shown in the following pictures:



2.12 Tree View



The tree view shows two root directories. The local hard disk (/) and the remote computers root (ssh/). The local root can be double clicked to open a standard file tree. Files can be opened via drag and drop into the render area, or by double clicking them with the left mouse button. This feature should work and needs no special treatment.

The remote connection needs some configuration beforehand to work.

Internally this functionality is based on "ssh" and "scp" to manage network connections so these tools must be present on the local system. On the remote system "cd" and "ls" must be present and ls has to be able to accept the switch "-l" for long view. Also it must be possible to start programs on the remote machine via ssh (try: ssh user@host ls -l).

Another thing is, that the specified user has to be able to log on to the specified host WITHOUT BEEING ASKED FOR A PASSWORD!

1. Create a key pair on the local machine, if there is none present, yet:
\$ ssh-keygen -t dsa -f ~/.ssh/id_dsa -C "you@example.com"
Generating DSA keys: Key generation complete.
Enter passphrase (empty for no passphrase): (If you use a passphrase you will need an ssh-agent! search the net...)
Enter same passphrase again:
Your identification has been saved in ~/.ssh/id_dsa

```
Your public key is:
1024 35 [really long string] you@example.com
Your public key has been saved in ~/.ssh/id_dsa.pub
$
```

2. Make sure the public key is in the ~/.ssh/authorized_keys file on the hosts you wish to connect to.
The following command appends your public key to the authorized_keys file on the remote host.
\$ cat ~/.ssh/id_dsa.pub | ssh you@other-host 'cat - >> ~/.ssh/authorized_keys'
you@other-host's password:
\$
3. Verify that DSA authentication works:
\$ ssh you@other-host
(Should not ask for a password and log on to "other-host".)

After this the connection should work. As with local files remote files can be dragged and dropped or double clicked. Note, that remote files are copied (via scp) to the local machine and stored in the "scp-tmp" directory of the qmga installation. The tree view window shows how much data is stored in this directory and offers a button to delete it all (when viewing a lot of rather large files the data will grow very fast...).

Also note, that files with the same name are overwritten in the tmp directory.