
Fos Documentation

Release 0.2

Eleftherios Garyfallidis & Stephan Gerhard

December 16, 2014

1	Actors	3
1.1	Static actors	5
1.2	Dynamic actors	5
2	Developers	7
3	Links	9
3.1	Polygon Mesh Processing	11
3.2	3/4 D Planetarium	11
3.3	Shaders	11
3.4	Engines	12
3.5	Wrappers	12
3.6	Web	12
3.7	Windowing API	12
3.8	Onscreen UIs	12
3.9	Debugging	12
3.10	Fonts	13
3.11	Video support	13
3.12	Scenegraph	13
3.13	Animation	13
3.14	Computational Geometry	13
3.15	Graph Visualization	13
3.16	GUI Backends	13
3.17	Physics library/engine	13
3.18	Computer Vision	14
3.19	Tutorials	14
3.20	OpenGL API	14
3.21	Documentations	14
3.22	Raytracer	14
3.23	CAD	14
3.24	3D Graphics Modeling	14
3.25	Biomedical	14
3.26	People	15
3.27	Conferences	15
3.28	Links	15
3.29	Scientific Visualization	15
4	Indices and tables	17

News: Working on major update of Fos. Stay alert :)))

Fos is a novel scientific visualization platform written in Python.

Fos provides

- a lightweight module for 3D renderings
- easy-to-use actors for presenting data
- clearly written and explained code
- a natural home for collaboration

Fos is based on Python, Numpy, Cython, OpenGL and PySide.

Find the source code [on GitHub](#)

Contents:

Actors

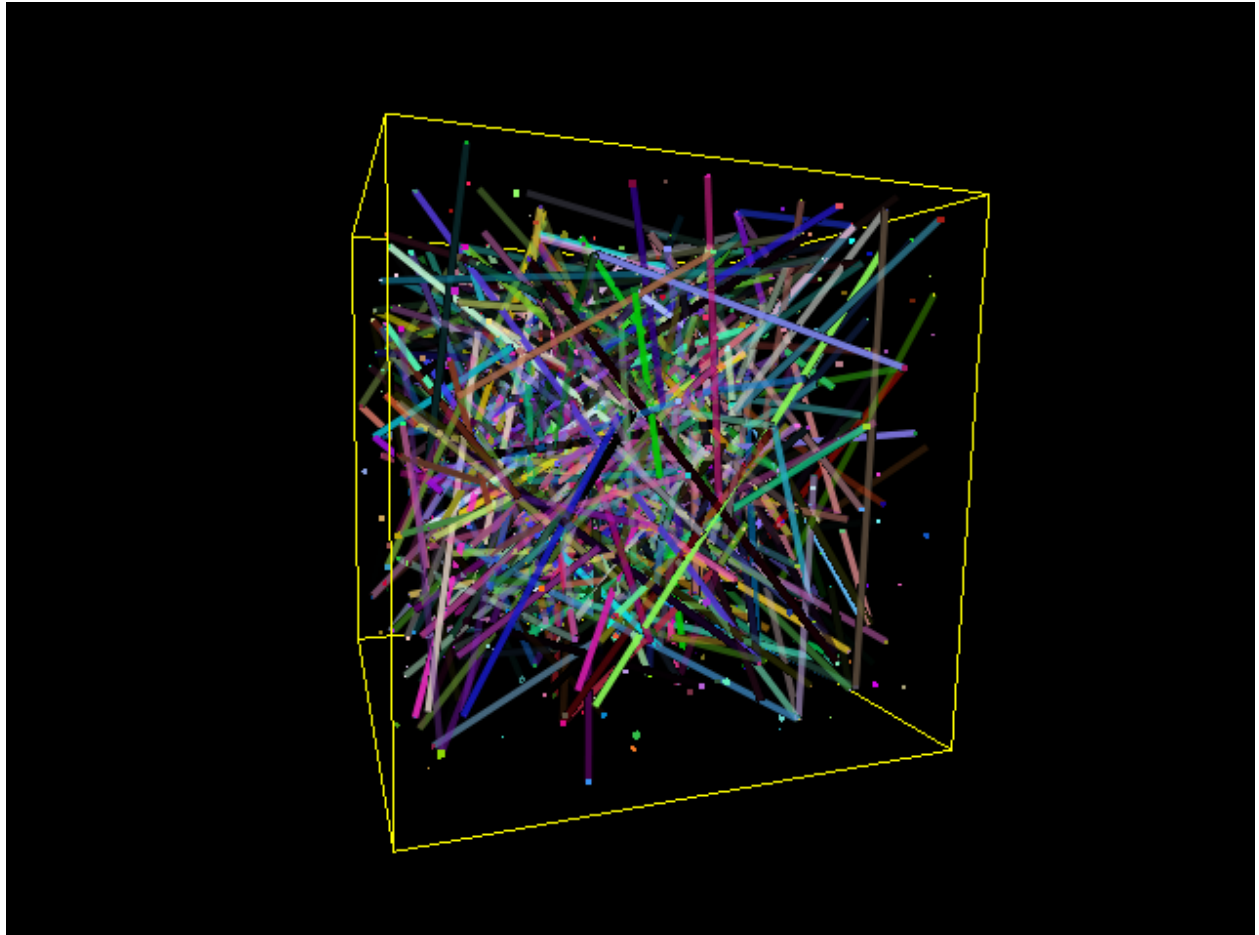
Fos includes various actors (visualization objects) for visualization. They expose an easy interface for creation, usually using Numpy arrays. Two classes of actors are defined:

Static actors For static structures that do not evolve in time.

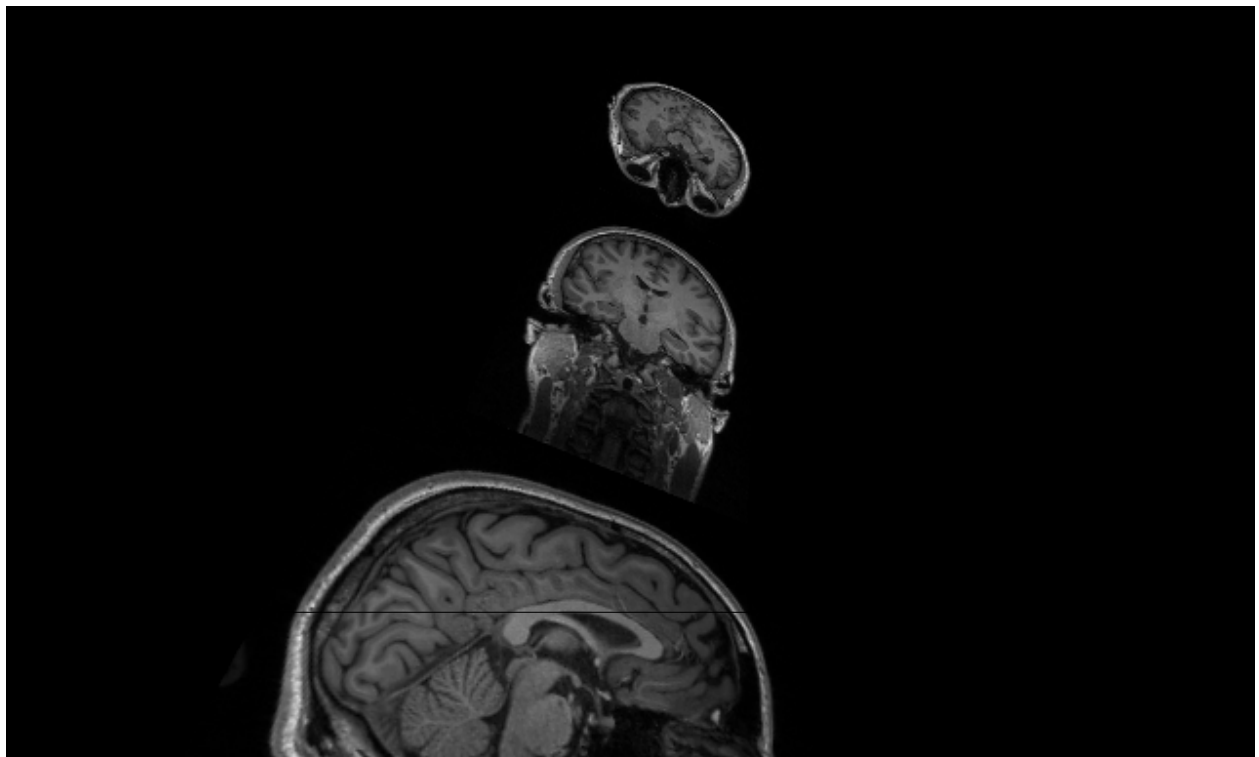
Dynamic actors One, or more parameters can have a time dimension, making the actor evolving dynamically.

1.1 Static actors

1.1.1 AttributeNetwork



1.1.2 ConnectedSlices



Developers

Eleftherios Garyfallidis

Stephan Gerhard

Ian Nimmo-Smith

Daniel Ginsburg

Vassilis Tsiaras

Links

Our aim is to collect knowledge, references and feature requests...

- also freeGLUT or OpenGLUT <http://www.lighthouse3d.com/cg-topics/glut-and-freeglut/>
- <http://www.panda3d.org/>
- <http://gamedev.stackexchange.com/>
- <http://www.sofa-framework.org/home>
- <http://www.pySide.org/docs/pySide/PySide/QtOpenGL/index.html>
- <http://neurobureau.projects.nitrc.org/BrainArt/Competition.html>
- <http://github.com/BenSmith/PythonOpenGLSuperBible4>
- gpu raycasting shader http://www.daimi.au.dk/~trier/?page_id=98
- <http://code.google.com/p/visvis/source/browse/visvisResources/raycasting.fragment.glsl>
- <http://jainek.de/projects/iBrain/>
- <http://vis.lbl.gov/Research/H5hut/>
- <http://www.songho.ca/opengl/index.html>
- http://3dshaders.com/home/index.php?option=com_weblinks&catid=14&Itemid=34
- <http://http.developer.nvidia.com/GPUGems2/>
- <http://pivy.coin3d.org/documentation/pycon>
- http://cfd.mace.manchester.ac.uk/sph/meetings/1stSPHERIC_workshop/Foggia_CSCS_SPHERIC_2006.pdf
- <http://personal.cscs.ch/~mvalle/visualization/TopVizProblems.html>
- http://www.ted.com/talks/anders_ynnerman_visualizing_the_medical_data_explosion.html
- nice sciviz: http://www.sciencevisuals.com/ourwork_projects.html
- more: <http://grahamj.com/>
- and more: <http://wehi.tv/>
- <http://sites.google.com/site/swinesmallpygletexamples/home>
- <http://kooneiform.wordpress.com/2010/01/05/pyglet-and-livecoding-1/>
- <http://code.google.com/p/pyglet-shaders/>
- <http://codeflow.org/entries/2010/dec/09/minecraft-like-rendering-experiments-in-opengl-4/>

- <http://yoshihitoyagi.com/projects/MasterProject/index.html>
- <http://lmi.bwh.harvard.edu/~banks/papers/illumBrain.pdf>
- <http://www.computer.org/portal/web/csd/doi/10.1109/TVCG.2007.70622>
- <http://alum.wpi.edu/~ginsburg/work.html>
- Vertex Array Objects http://www.opengl.org/wiki/Vertex_Array_Object
- nice ideas codeflow.org
- animation and general design ideas for fos <http://codeflow.org/entries/2010/nov/29/verlet-collision-with-impulse-preservation/>
- GPU gems: http://developer.nvidia.com/object/gpu_gems_2_home.html
- gletools, lots of things to learn: <http://codeflow.org/entries/2009/jul/31/gletools-advanced-pyglut-utilities/>
- interactorium http://www.systemsbiology.org.au/downloads_interactorium.html
- tutorial on opengl vbo <http://nehe.gamedev.net/data/lessons/lesson.asp?lesson=45>
- <http://www.informit.com/articles/article.aspx?p=461848&seqNum=2>
- <http://github.com/matthew-brett/miniglumpy>
- pyffmpeg
- pymt packages: <http://pymt.eu/wiki/Devel/FeaturesByLibraries>
- read: http://groups.google.com/group/pyglut-users/browse_thread/thread/ab1d988f2b5373cf
- see `glumpy4bounding box`, e.g. `function in box`
- <http://local.wasp.uwa.edu.au/~pbourke/geometry/polygonise/>
- <http://code.google.com/p/pythonisosurfaces/>
- <http://stackoverflow.com/questions/145838/benefits-of-inline-functions-in-c>
- overview of opengl draw calls: <http://tartley.com/?p=1045>
- pyglut-shaders: <http://code.google.com/p/pyglut-shaders/>
- maybe helpful for translucent textures: <http://compileandrun.com/?p=39>
- <http://code.astraw.com/projects/motmot/pyglutarrayimage.html>
- checkout this: <http://invizian.loni.ucla.edu/>
- <http://www.mevisslab.de/>
- conference: <http://vis.computer.org/VisWeek2010/>
- Interactive Pyglut usage with cocos2d http://www.examplelab.com.ar/python/interactive_cocos2d/python_interactive_session_with
- MedX3D: Standards enabled medical 3D <http://www.web3d.org/x3d/workgroups/medical/>
- <http://cgkit.sourceforge.net/doc2/index.html>
- <http://www.h3dapi.org/>
- SciGL (Scientific OpenGL Visualization ToolKit) <http://www.loria.fr/~rougier/coding/scigl/index.html>
- <http://www.loria.fr/~rougier/coding/index.html>
- In a similar vain, but dead now. Why? <http://www.opendx.org/>
- SVG for pyglut <http://www.supereffective.org/pages/Squirtle-SVG-Library>

- The Vish Project <http://vish.origo.ethz.ch/>
- YT <http://yt.enzotools.org/doc/index.html>
- http://yt.enzotools.org/video_gallery.html
- http://yt.enzotools.org/doc/extensions/volume_rendering.html
- VISIT <https://wci.llnl.gov/codes/visit/home.html>
- <http://cs.nyu.edu/~weinkauf/publications/absbenger10a.html>
- 3/4 D Planetarium

<http://www.haydenplanetarium.org/universe/download/> http://www.ted.com/talks/carter_emmart_demos_a_3d_atlas_of_the_universe.ht
<http://virdir.ncsa.illinois.edu/partiview/>

- Celestia <http://www.shatters.net/celestia/>
- Gephi discussion about 3D in web: <http://forum.gephi.org/viewtopic.php?f=9&t=89>
- web image authoring tips: <http://www.web3d.org/x3d/content/examples/X3dSceneAuthoringHints.html#Images>
- Gary Flake: is Pivot a turning point for web exploration? http://www.ted.com/talks/gary_flake_is_pivot_a_turning_point_for_web
- fiber bundle, F5 format! <http://www.fiberbundle.net/>
- Meminfo: <http://www.geeks3d.com/20100531/programming-tips-how-to-know-the-graphics-memory-size-and-usage-in-opengl/> (GL_NVX_gpu_memory_info or WGL_AMD_gpu_association)

3.1 Polygon Mesh Processing

Geometry processing: surface reconstruction from point samples, filtering operations for noise removal, geometry analysis, shape simplification, geometric modelling. queries: e.g. p e R3. Is it inside or outside? distance from p to surface <http://cgal.org> <http://openmesh.org> <http://meshlab.org>

3.2 3/4 D Planetarium

<http://www.haydenplanetarium.org/universe/download/> http://www.ted.com/talks/carter_emmart_demos_a_3d_atlas_of_the_universe.ht
<http://virdir.ncsa.illinois.edu/partiview/>

3.3 Shaders

- <http://swiftcoder.wordpress.com/2008/12/19/simple-glsl-wrapper-for-pyglet/>
- GLSL and Python <http://www.pythonstuff.org/glsl/index.html>
- GLSL Tutorial <http://www.lighthouse3d.com/tutorials/glsl-tutorial/>
- <http://www.slidefinder.net/S/Shader/11893836/p2>
- <http://gitorious.org/personal-julian-ibarz/obj-viewer/blobs/master/shaders/cullDirection.geom>
- Shader-Based Wireframe Drawing http://cgg-journal.com/2008-2/06/index.html#tth_sEc2.2
- <http://www.flashbang.se/archives/307>

3.4 Engines

- `pyglet`
- `pygame`
- 3d cloud game engine - <http://www.pysoy.org/>
- soya3d - <http://home.gna.org/oomadness/en/soya3d/index.html>
- <http://www.panda3d.org/>
- `visvis`

3.5 Wrappers

- <http://pyopengl.sourceforge.net/context/>

3.6 Web

- `x3dom` : http://www.x3dom.org/?page_id=5
- `web3d` : <http://www.web3d.org/about/overview/>
- (<http://www.smartclient.com/smartgwt/showcase/#main>)
- `WebGLU` : <http://blog.one-geek.com/>
- `X3D and HTML5 Summary` : http://www.web3d.org/x3d/wiki/index.php/X3D_and_HTML5_Summary
- `Examples` : <http://www.web3d.org/x3d/content/examples/Basic/>
- `Processing.js` : <http://processingjs.org/exhibition>

3.7 Windowing API

- `glut` (with `freeglut3` implementation)
- `sdl` (used by `pygame`)

3.8 Onscreen UIs

1. `simplui` <http://code.google.com/p/simplui/>
2. `kytten-read-only` <http://code.google.com/p/kytten/>
3. <http://www.antisphere.com/Wiki/tools:anttweakbar> (C++)

3.9 Debugging

- <http://valgrind.org/>

3.10 Fonts

1. pyftgl (needs ubuntu libftgl2, ftgl-dev)

3.11 Video support

1. pyglet, needs libavbin0

3.12 Scenegraph

- <http://github.com/mshaw/OOgler.git> (need python-zope.component)
- <http://www.opensg.org/>

3.13 Animation

<http://github.com/tuomasj/pyglet-animationsprite>

3.14 Computational Geometry

- www.cgal.org
- ghull
- Applied Geometry Group ETH :<http://www.agg.ethz.ch/>
- bounding volume code <http://copythatfloppy.net/bzr/xyzyz/>

3.15 Graph Visualization

- GVSR <http://gvsr.polytech.univ-nantes.fr/GVSR>

3.16 GUI Backends

- possible to include pyglet in wx (see ML)
- see visvis backends

3.17 Physics library/engine

- <http://code.google.com/p/pymunk/>
- collision.pyx
- <http://www.magic.ubc.ca/artisynth/pmwiki.php?n=Demo.Demo>

3.18 Computer Vision

- PyOpenCV <http://code.google.com/p/pyopencv/>

3.19 Tutorials

- <http://nehe.gamedev.net/>
- http://pyopengl.sourceforge.net/context/tutorials/shader_1.xhtml
- http://pyopengl.sourceforge.net/context/tutorials/shader_3.xhtml

3.20 OpenGL API

- <http://tartley.com/?p=1045>

3.21 Documentations

- <http://pyopengl.sourceforge.net/documentation/index.html>
- <http://www.pyglet.org/documentation.html>

3.22 Raytracer

- yafaray
- renderman
- povray

3.23 CAD

- FreeCAD based OpenCascade
- Wings3D

3.24 3D Graphics Modeling

- k3d <http://www.k-3d.org/>
- blender3d

3.25 Biomedical

- <http://www.sci.utah.edu/cibc/software/107-map3d.html>

3.26 People

- Gordon Kindlemann, teem nrrd <http://lmi.bwh.harvard.edu/~gk/> <http://people.cs.uchicago.edu/~glk/>
- cool ideas, bumptop http://www.ted.com/talks/anand_agarawala_demos_his_bumptop_desktop.html
- Almar Klein <http://code.google.com/p/visvis/>
- Jonathan Hartley <http://tartley.com/>
- Werner Benger <http://www.cct.lsu.edu/~werner/>
- Nicolas Rouger <http://code.google.com/p/glumpy/>

3.27 Conferences

- <http://vis.computer.org/VisWeek2010/>

3.28 Links

- Illuminated streamlines: http://www.scivis.ethz.ch/research/projects/illuminated_streamlines

3.29 Scientific Visualization

- SciVis Course ETH Zurich: http://www.scivis.ethz.ch/education/scivis_course/notes
- SciVis ETH Zurich: <http://www.scivis.ethz.ch>

Indices and tables

- *genindex*
- *modindex*
- *search*