Advanced features

A small input on what else is possible

Outline

- Display results
- File Management
- Geometry and Math
- Code support
- Execution interface
- VRskeletonization

Display result

- Video generator
- Export image
- Plot

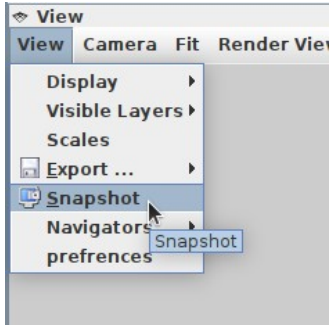
Video generation

- Integrated panel:
  Panels > *Video generator*
- Requires FFMPEG installed
  on the computer

https://wiki.grogra.de/doku.php?id=tutorials:video-plugin

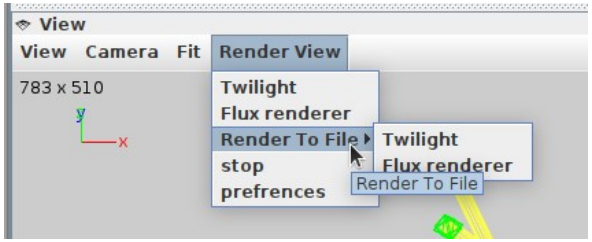
Export image

Snapshot of the current 3d view




```
public void snaps ()
{
     makeSnapshot(getWD() + "test.png");
}
```

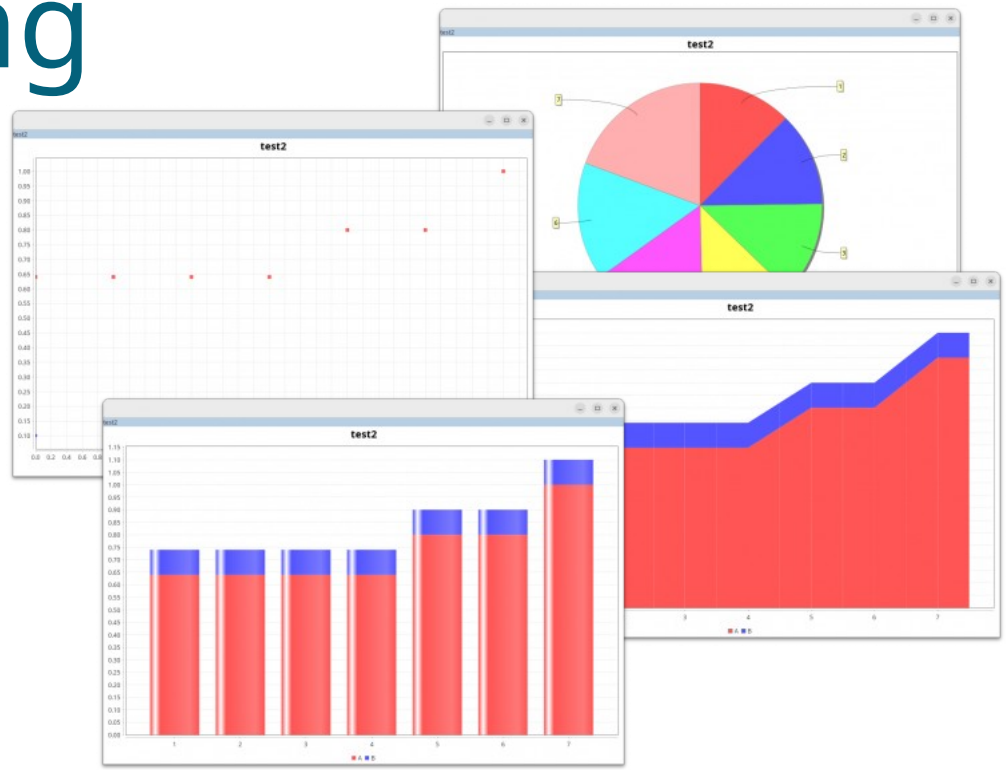
Render in file




```
public void rend ()
{
     // render with the twilight renderer
     makeRenderedImage(getWD() + "rendered.png");

     // render with GPUFlux renderer
     makeRenderedImageFlux(getWD() + "fluxrendered.png");
}
```

Datasets & Plotting

- GroIMP includes datasets (like simple spreadsheets)
- Datasets can be plotted in automatically updated charts



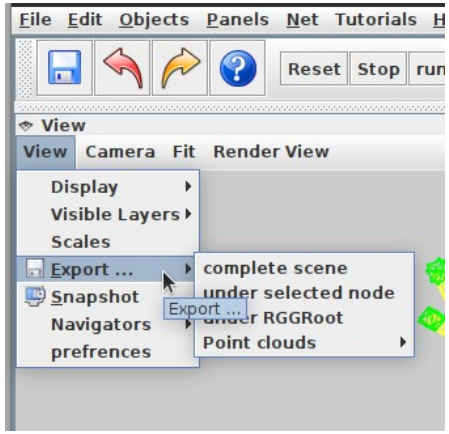
https://wiki.grogra.de/doku.php?id=tutorials:plotting-data

File management

- Export
- Import
- Object explorer

Exporting graph data

Using the GUI



Using XL code

```
public void exportgraph ()
{
    // Export3Dscene can support many set of arguments
    // export the complete scene as obj to the file myexport.obj
    export3DScene(getWD() + "myexport.obj", "obj");

    // export the complete scene but only nodes that are currently in a "visible layer"
    export3DScene(getWD() + "exportvisible.obj", "obj", true);

    // export under node with id 25
    export3DScene(getWD() + "exportspecific.obj", "obj", (Node)graph().getNodeForId((long)25) );
}
```

https://wiki.grogra.de/doku.php?id=tutorials:export-object-from-groimp

Exporting graph

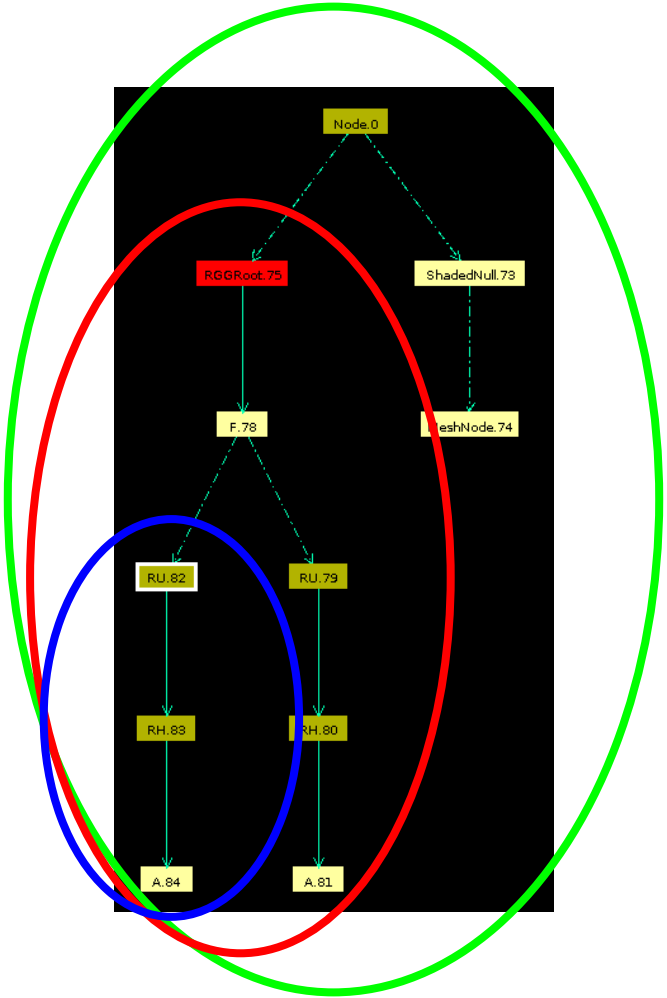
Export either as:

- Geometry (obj, ply, …)
- Graph (rsml, qsm, …)

Complete scene

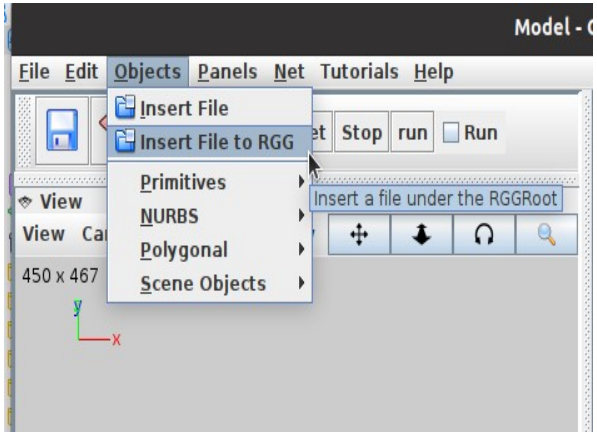
Under RGGRoot

Under selected



Importing data

Using the GUI



Using XL code

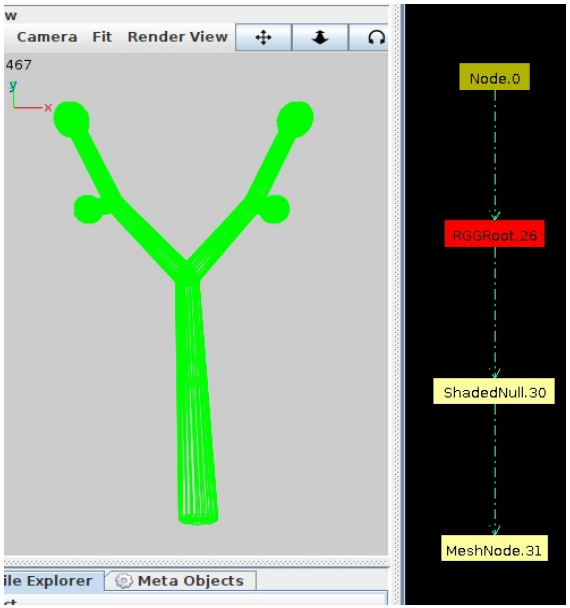
```
public void importobj ()
{
    // Import the file under the Graph root
    importNodeFromFile(getWD() + "test.obj", "image/vnd.obj");

    // Load the node as a Variable. which can be added in the graph if wanted, or used in queries, ...
    Node n = loadNodeFromFile(getWD() + "test.obj", "image/vnd.obj");
}
```

https://wiki.grogra.de/doku.php?id=tutorials:import-object-in-groimp

Importing data - Position

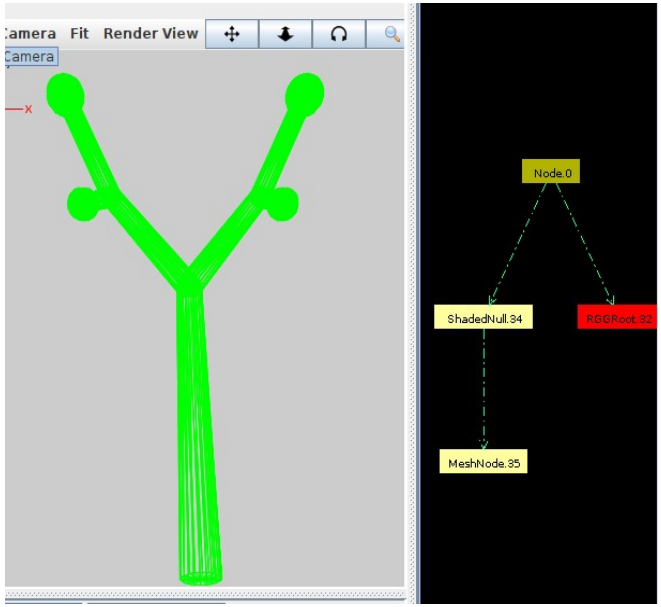
Under RGGRoot

Deleted when project is “reset” or “recompiled”



Under Graph root

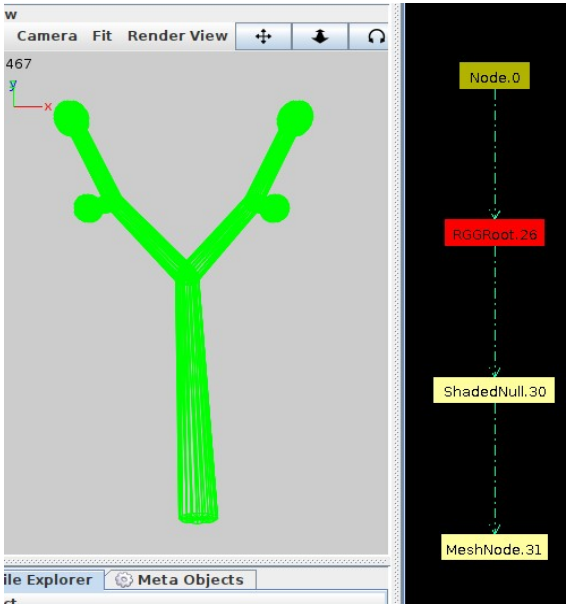
Persistent import



Importing data - Formats

Object import (obj, ply, …)

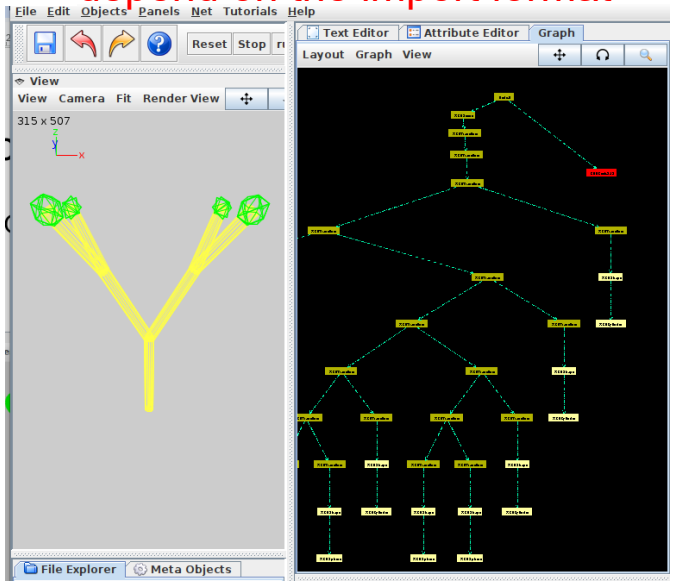
⇨ Only includes geometry



Graph import (x3d, gltf, qsm, …)

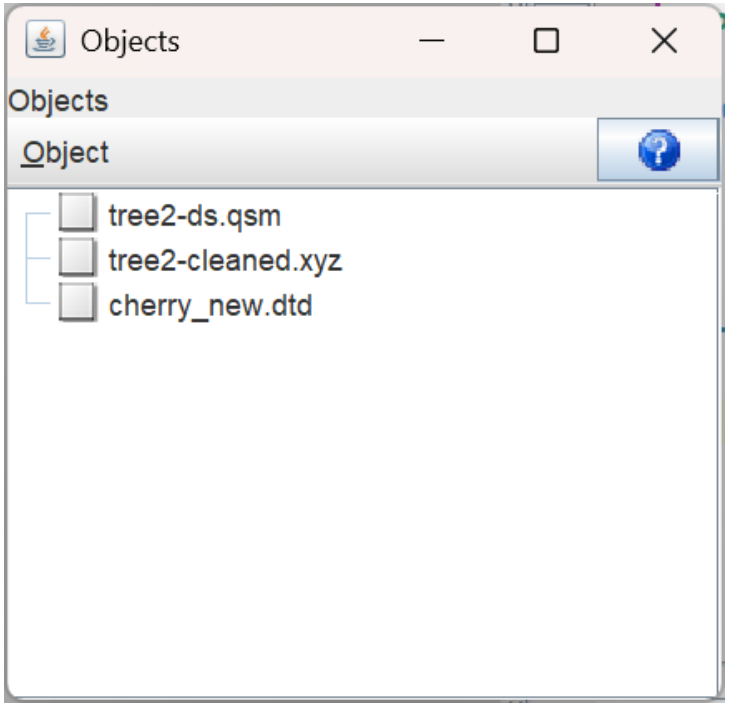
⇨ Geometry & topology

⚠ The graph structure can depend on the import format



Object explorer

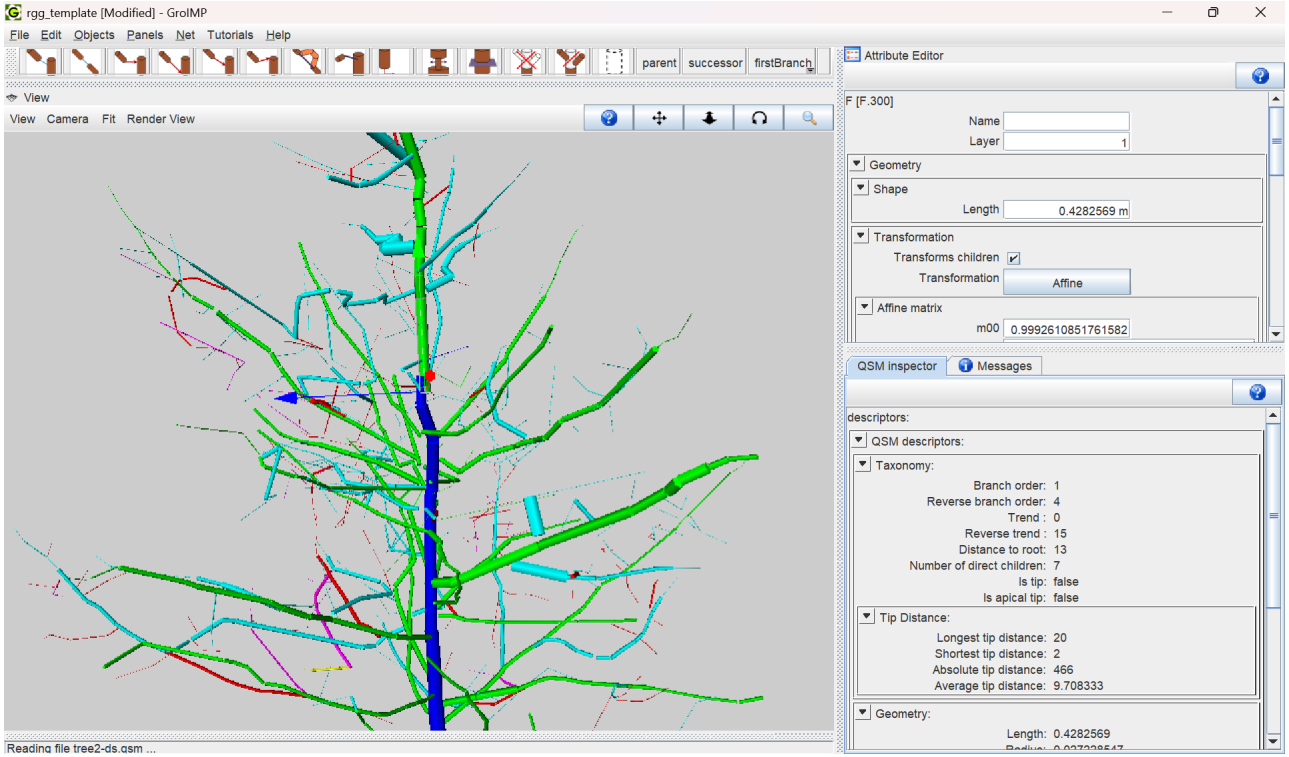
- Importable files can be added as resources
- Objects can then be referenced similar to shaders etc.



https://wiki.grogra.de/doku.php?id=tutorials:import-object-in-groimp#using_the_object_explorer

QSM integration

- Quantitative structure models
- Import and export
- Filtering and Editing
- Additional descriptors



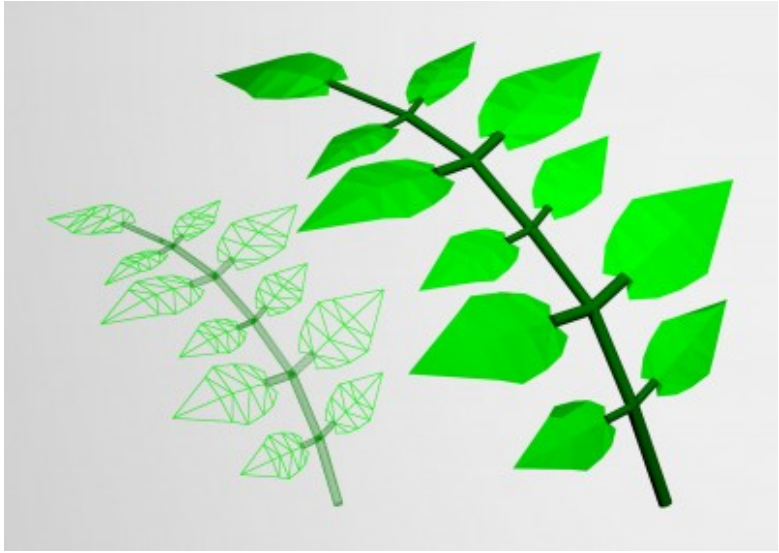
https://manual.grogra.de/de.grogra.qsm + tutorials

Geometry and Math

- Triangulation
- Convex Hull & Alpha shape
- Geometrical analytics
- ODE

Triangulation

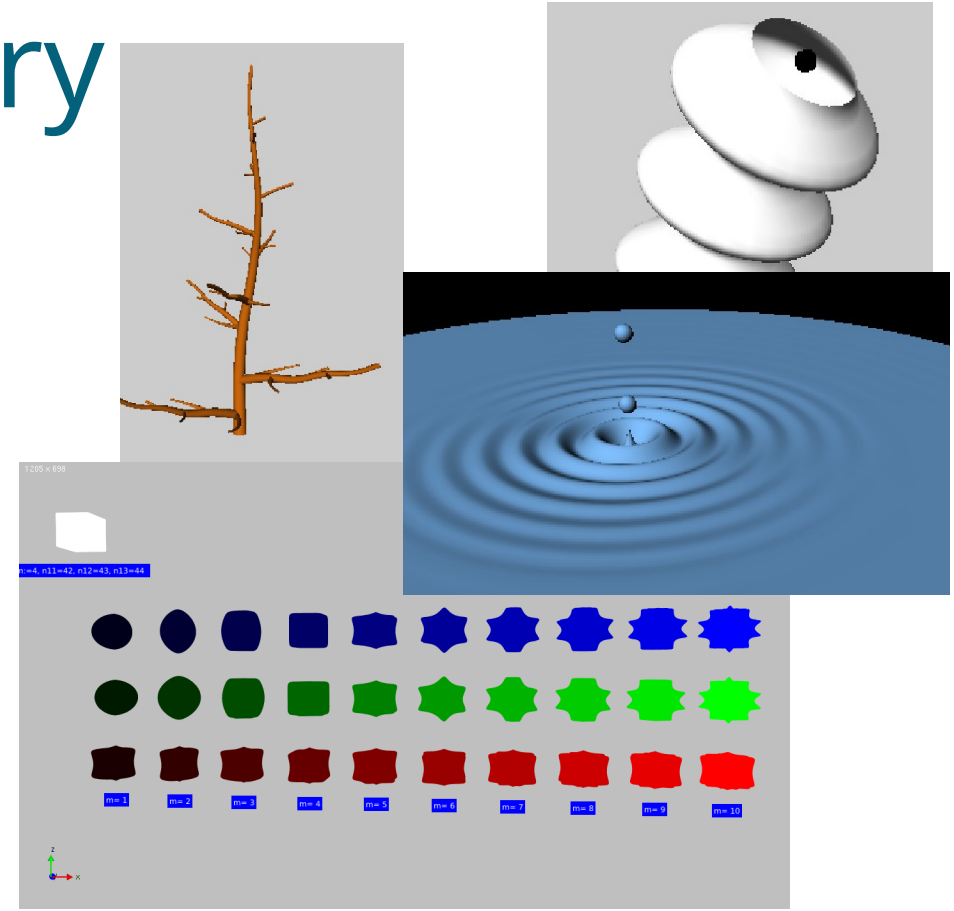
- Creating 3d objects by describing individual triangles



https://wiki.grogra.de/doku.php?id=tutorials:leaf-triangulation

Advanced geometry

- Super shapes
- Nurbs
- You can find examples on the gallery and embedded



Geometrical analytics

- GroIMP provides a collection of functions to analysis geometrical properties of Objects.
- e.g.
  - Location & EndLocation objects in global coordinates
  - Angel between two Obejcts
  - ConvexHull or Alpha shape of a collections of points
- You can find this in the embedded examples and the function Browser

ODE

- GroIMP includes a framework for solving ordinary differential equations

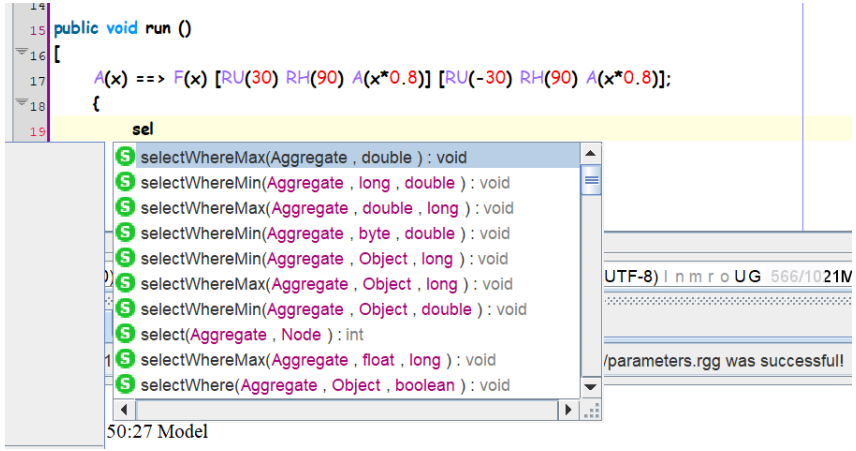
https://wiki.grogra.de/doku.php?id=tutorials:ode-framework

Coding- support

- Auto completion
- Code outline
- Project wide search

Auto completion

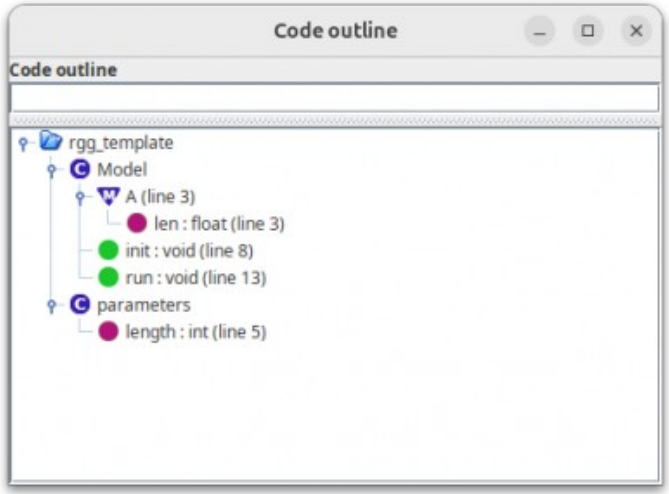
- GroIMP supports two types of auto completion
  - Static
  - Scope based
- By default non is selected



https://wiki.grogra.de/doku.php?id=additional-functionality:autocomplete

Code outline

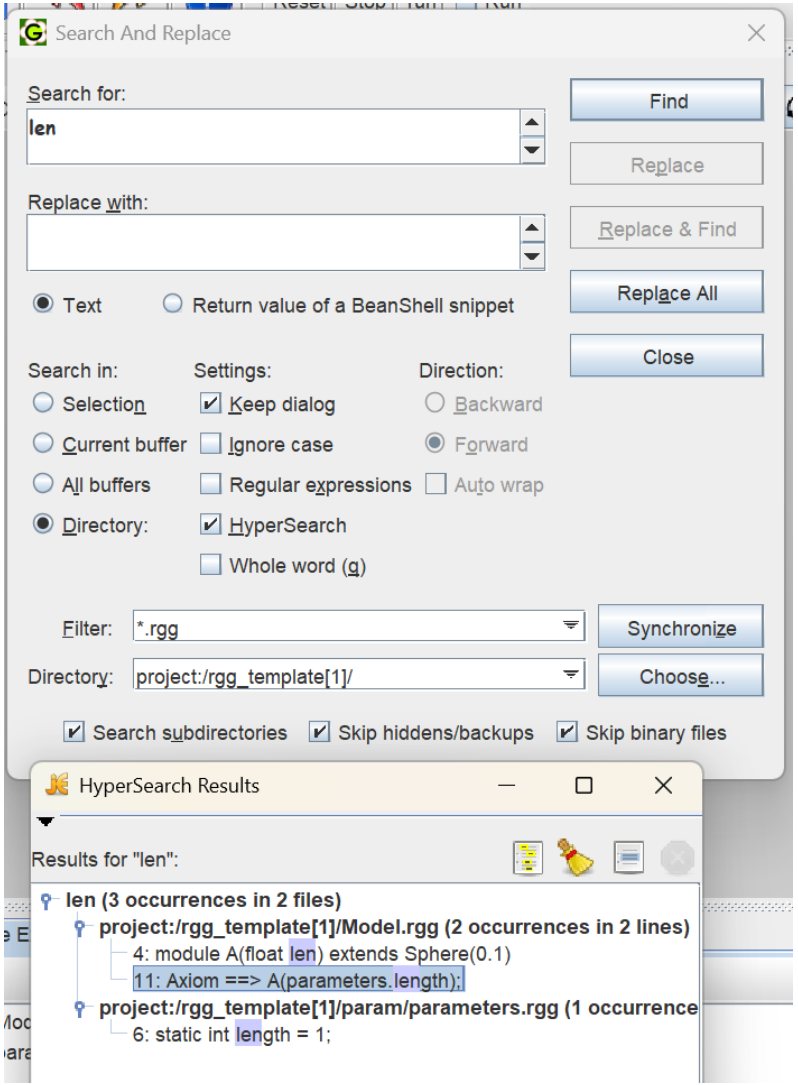
- Code outline is basically a linked table of contents for your code
- You can find it on *panels* > *Code outline*



https://wiki.grogra.de/doku.php?id=user-guide:more_panels#code_outline

Full project search

- Jedit allows to search all files in a project
- Just change to the “search in” option: “Directory”
- There is also an additional project wide file search for all types of files(*Panels* > *Search project*)



Execution interface

- Headless mode
- API

Headless

- GroIMP can be started without its graphical interface, in a non interactive headless mode.
- This can be used to run a model or a command without interaction
- Useful for execution on non GUI environments or for automation

https://wiki.grogra.de/doku.php?id=user-guide:additional_interfaces:headless

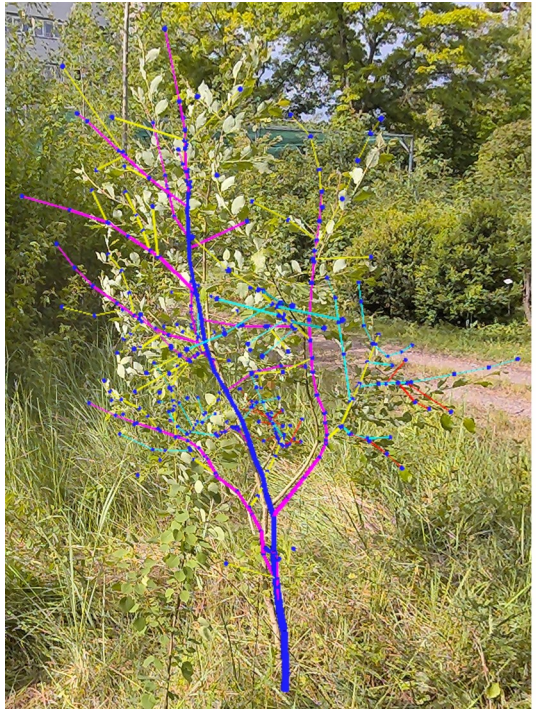
API(GroLink)

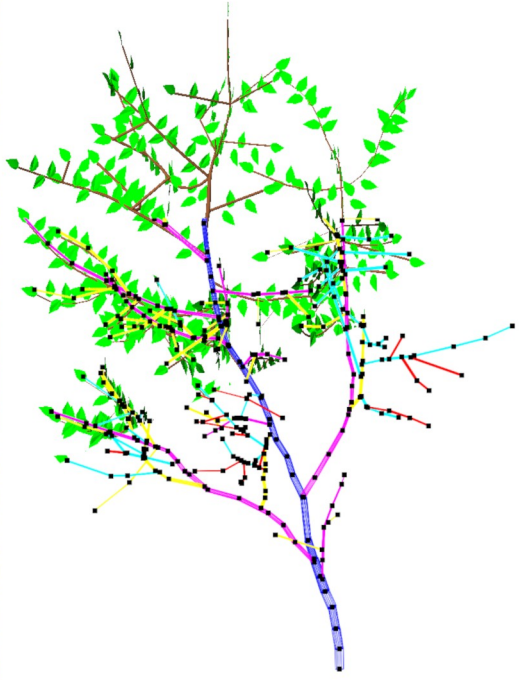
- A statelss http api
- Allows to access GroIMP form other software/projects
- Works with any software that can send http request
- A Python and a R library are provided to ease usage

https://wiki.grogra.de/doku.php?id=user-guide:additional_interfaces:api

VRskeletonization

- Linking GroIMP to a VR headset





- What we are going to present at the PMA ;)