

# Statistical Geometry Processing

## Winter Term 2011/2012

### Assignment Sheet #0: Installing and Using GeoX

**Author:** Michael Wand  
**Contact:** [mwand@mpi-inf.mpg.de](mailto:mwand@mpi-inf.mpg.de)

**Handed out:** Oct. 25th 2011  
**Due:** *no due date*

#### Remark

This is an *optional assignment sheet* that is not graded in any way. It is meant to introduce the GeoX framework and acquire some practical experience with it. We will discuss this in more detail in the upcoming lab session (November 2nd) during the tutorials (Wednesday 17:15-18:45h, Room 021). Do not hesitate to contact us if there are any questions on how to use GeoX or QT.

#### Assignment 0.1: Setup the system

Download the GeoX zip archive from the lecture's home page. Get a Linux or Windows box and install the latest version of QT, if necessary (Versions > 4.2.x should work). Setup your development environment following the instructions in the PDF document provided ("Using the GeoX Framework").

#### Assignment 0.2: Working with viewers and geometry

**(a)** Create a new experiment based on the 2D viewer (see "ExampleExperiment2DGraphics.h/cpp" for an example). The viewer is selected by overriding the inherited "createView" method.

**(b)** Implement and register a method that creates some geometric objects: A circle, a regular polygon, or something the like. Methods can be mapped to buttons in the GUI by registering them. Currently, only parameterless methods are supported. You should use the macro "ADD\_NOARGS\_METHOD()" as demonstrated in the Examples. Parameters need to be stored in the experiment object itself.

**(c)** Settings: introduce parameters for your objects within the experiment object (such as a member defining the radius of the circle to be created). Member variables can be registered for interactive modification in the GUI using the "ADD\_XXX\_PROP()" macros, where XXX refers to the data type (see "Using the GeoX Framework"). Examples are given in "ExampleExperiment.h/cpp". You can also load and save your experiments by automatic serialization (try this out and check if it works; pay attention to versioning).

**(d)** Now create a new experiment that uses the 3D viewer. Again, create some geometric primitives, such as a triangle mesh approximation of a sphere.

#### Assignment 0.3: Build your own viewer

This assignment is more involved: You should now build your own custom viewer. This requires some basic knowledge of QT. Viewers are subclasses of `QWidget` and use inherited methods as well as signals and slots to communicate with the QT library and the rest of the application. For more details, see <http://doc.qt.nokia.com/> and/or talk to your instructor/TAs.

**(a)** Build a new viewer that can display images. It should provide methods to create a new view of an image (for example, by a member method "int addImage(QImage\*)") and return an identifier to access the image later. There should be a rudimentary user interface to switch between different the images.

**(b)** Use QT's mouse events to implement a simple drawing application where you can directly paint on the images. You can integrate the functionality either in the viewer or in a separate experiment.