1. Human aspects of the proposal

My name is Oana Niculaescu and I am a 4th year student at the Politehnica University of Bucharest - Romania, studying Computer Engineering. I am a science enthusiast and I try to learn as many new things as I can but at the same time I try to become proficient at the things I already know.

During my studies I've become familiar with C and C++, most of the projects I've worked on used one of those two languages, Java and Python. Those are the languages that I think are relevant for this project and your organization to know about. I have used Git and SVN before on a daily basis, so I am more than familiar with it.

I have not contributed to Scilab before, but I would love to continue to contribute after SOCIS to the project if the mentors consider that I can bring value to it. Although, I haven't used Scilab before, I have used Matlab and Octave, that I think are pretty similar to this software.

A small portfolio with samples of my work can be found on my Github account [1] , also on there can be found my updated CV [2] – here some more information about me can be found (phone number, address).

I can be reached using the following:
e-mail: oana.niculaescu@gmail.com
IRC: oana/oana_
skype: oana.niculaescu

2. Technical aspects of the proposal

I am proposing the implementation of the X3D export feature in the Scilab project.

X3D is an open software standard for defining and communicating 3D content. X3D is a file format specification with requirements on how that file is to be displayed. It is encoded in XML format so it can be easily extensible and edited and the 3D objects can be manipulated using C/C++ and Java.

A X3D scene is described by the contents of a file that has a common XML structure:

- a file header – XML header
- an X3D root node, that includes version and profile attributes
- a head section with components and meta statements (both optional)
- the X3D graph scene and its child nodes
- the end of the root node

What kind of features does X3D support:

- atoms (as spheres), bonds (as cylinders)
  - including atom size, bond thickness, color
- drawn objects (lines, arrows, triangles etc)
- isosurfaces
- labels – text, color, font family and font style
- dots, halos, stars, polyhedra, vectors, dipoles, ellipsoids
- NurbsCurves, PolyLines, shapes (simple shapes, pure simple shape, heightfield shape)

What is not supported/not supported well:

- the orientation and zoom are not always kept
- labels, 3D-positioned and font size and offset for them

What I am proposing is the implementation of a new exporting module, that will contain for the moment the X3D export option and will be in the future extended to contain other exporting formats

(.dae format for the Collada open standard, VTK format). The module will be similar to the graphic_export module already implemented in the Scilab source code. I saw that there is used JNI for calling functionalities from the Java classes that are actually implementing the exporting to the BMP file format, but I don't think that will be necessarily in this case. I am planning to use SCEW[3] (Simple C Expat Wrapper) to be able to write the XML files. Expat[4] is an XML parser library written in C. It is a stream-oriented parser in which an application registers handlers for things the parser might find in the XML document. SCEW facilitates the use of Expat providing functions to load and access XML elements without the need to create Expat event handling routines every time you want to load a new XML document. SCEW uses a DOM like XML object model for parsing the files and objects are saved in nodes. SCEW I/O system is based on Readers, Writers and Printer. A reader is a common mechanism to load data from different sources (files, memory, ...). A common mechanism means that the functions to read data, for example, from a file or from a memory buffer, are the same. SCEW writers follows the same idea behind the readers, that is, common routines are used to write data to any kind of sources (files, memory, ...). The SCEW printer provides routines to write SCEW XML data (trees, elements and attributes) to a SCEW writer.
The new feature will be available to the user using the command line in Scilab, calling it similarly with the export to BMP file format, something like this:
→ xs2_x3d(graphic_window_number, 'output_file.x3d')

I am going to present a rough draft of the schedule that, if I will be chosen to participate in this edition of SOCIS, must be refined as soon as possible with the direct mentor.

Weeks 1-2:

- setting up a new Scilab module
- gaining more understanding on how to actually use the SCEW library and setting it up in the Scilab project

Week 3-4:

- exporting for starters a small set of features to X3D
- I propose I will start with polyline, disks and contours in 2D and writing tests for them

Week 5-7:

- exporting other 2D shapes (this must be discussed with the mentor in order to establish which other shapes are important to be implemented)
- writing test cases for those too
- starting the export of some 3D shapes – sphere, cylinder, cone, box, heightfield shapes and nurbs
- writing test cases for those shapes

Week 8-10:

- implement the material texture, transparency and zoom ratio for the objects that will be exported
- I am thinking that the user can call the function using other parameters too, e.g.:
    - xs2_x3d(graphic_window_number, transparency_index, zoom_index, 'output_file.x3d')
    - the transparency_index is a number between 0-100, with 0 being very transparent and 100 being opaque
    - the zoom_index is a number that indicates how much the size of the object in the saved file will grow comparing to the initial object, it will be a number between 0-100, and it will be

added to the initial size of the object, for example xs2_x3d(0, 100, 10, 'output.x3d') will save the object in graphic window #0 with an opaque transparency and a size of 110% to the output.x3d file

  - the transparency and zoom index will be optional

Week 11-12:

- writing help pages for the implemented functions
- writing documentation
- time left for any un-forseeable problems that might appear during the coding period

Deliverables:

- a new Scilab module that will implement the export of files to the X3D format
- a set of tests that will prove the functionality of the new module
- documentation about the new implemented feature

[1] https://github.com/elf11
[2] https://github.com/elf11/CV/blob/master/Oana_Niculaescu_CV.pdf
[3] https://github.com/aconchillo/scew
[4] http://expat.sourceforge.net/