TD2 : display and navigation

1 Preliminary note

A `skeleton` is provided in order to have a basic structure to start working with. However, it is incomplete and needs introduction of new variables and new methods in order to produce satisfying results. It is perfectly allowed to add news files and classes, it is also adviced to import some of the code you wrote during the first assignment.

The `skeleton` can be built with the following command from `skeleton` folder :

```
mkdir build && cd build && qmake --qt=qt5 .. && make
```

This will produce a binary named `faces_viewer` which opens a `Qt` window containing an empty `OpenGL` Widget.

Most of the tasks presented in this assignment can be implemented separately.

2 Display and basic interaction

The initial version of the `skeleton` loads a `json` file containing faces to display, but it does not implement drawing inside the `GLWidget`.

The first task is to start displaying the `FaceCollection` which is loaded by the program at the beginning. This requires to implement several elements in `glwidget.cpp` :

— Initialization of the OpenGL ressources in `initializeGL`

— Definition of the point of view in `paintGL`, initially, the object should be centered in the `GLWidget`

— Drawing of the faces in `paintGL`

A button connected to the `ViewerWidget::loadFile` slot is initialized in the skeleton, however the dialog to request the file is not implemented. In order to change dynamically the model viewed, you need to implement this method.

In order to be able to view the inner part of the object, it is mandatory to use transparency for the faces, and to include an alpha channel. For this assignment you should use the same transparency value for all the triangles and it should be possible to modify by changing a value on a `QSlider`.

You should ensure that it is possible to resize the window and that it does not affect the aspect of the object shown in your window.

3 Moving the camera

In order to allow users to change the point of view, you should implement a mouse-based user interface with the following elements :

— Holding the left button of the mouse pressed and moving the mouse allows to rotate the point of view. You should make sure that there are no singularities in the rotation.

— Holding the right button of the mouse pressed and moving it should allow translation of the point of view.

— Using the mousewheel should allow to zoom-in and zoom-out.

For all the previously mentioned motions, the speed should be highly increased when the key `SHIFT` is pressed.

4 Evaluation

Implementation of all the features presented in this assignment will be tested to establish your grade.

— The result should be sent before the deadline (according to website).
— For the archive and the title of the mail, names should be ordered alphabetically
— The title of the mail should be `[4TTV904U] TD2: Name1 Name2 XXX`
— The mail should have all members of the group in CC
— A file named `name1_name2_XXX.tar.gz` should be attached
  — Running `tar -xzf name1_name2_XXX.tar.gz` should not throw any error.
  — The archive should contain a folder named `name1_name2_XXX` which should contain only the necessary elements to compile the project.
  — Running `mkdir build && cd build && qmake -qt=qt5 .. && make` should succesfully build the project