TD4: Opening STL and OBJ files

1 Introduction

This assignment reuse previous code and focus on using classical format used to store 3d objects: `STL` and `OBJ`.

The aim is to increase the range of the files supported and to understand how other format can work. Once the assignment is finished, it should be possible to load `PGM3D`, `Json`, `STL` and `OBJ` files using your program.

Note: when converting strings to floating points numbers, you might use standard function such as `std::stod`. In this case, be careful about the values of the locale (you can display it using `locale`), the default values for `LC_NUMERIC` on the computer of the CREMI is french, which leads to consider that the separator for decimal numbers is a comma. In case the numbers are rounded, you can temporarily override the locale using: `LC_NUMERIC=C build/faces_viewer ...`

2 Support for ASCII STL files

The `STL` format allows to describe the geometry of a 3 dimensional surface by decomposing it into triangles. This simple format do not contain information about the color nor the texture of the object. For this assignment, support for `ASCII STL` is enough, you do not need to implement reading from `binary STL`.

Since `STL` does not contain information about the color, you can simply consider that all the faces have a color of 1.0.

A few example files are provided in the assignment archive, it is normal that your program might not be fluid when using some of them due to the large number of faces they contain, especially `robot_low.stl`.

3 Partial support for OBJ files

The `OBJ` format is much richer than `STL`, it allows to specify textures, colors and faces with more than 3 or 4 vertices. For this assignment, you will only be requested to obtain the geometrical structure from the file, not the color neither the texture.

You can find simple `OBJ` files with preview examples on `https://people.sc.fsu.edu/~jburkardt/data/obj/obj.html`. In some cases, the normal of the faces is not specified. To support those files, you will probably need to modify the class `Face` to take into account that normal might not be specified.

In order to be able to filter faces based on their normal, you can take a look at `FACE_CULLING`[1].

[1]`https://www.khronos.org/opengl/wiki/Face_Culling`

4 Evaluation

Since this assignment is mainly based on supporting the reading of files in multiple formats, evaluation will consider error handling. Recommended behavior on invalid files is to open an error dialog window with an explicit error.

While it is not necessary to have all the features from assignment 1 to 3 operational for this assignment, basic display of faces and camera navigation have to be functional.

- 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] TD4: 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