



Lab Project 3. The Scene Viewer

In this lab session, you will finish the 3D scene viewer software.

Class Scene: Implements a scene. This class inherits from `SceneInterface` and implements its methods. The scene contains:

- A pointer to the GUI.
- A camera (class `Camera`).
- A list of 3D objects (class `Objec3D`).

The class must, at least, define the following methods:

- `draw() const`
Draws all objects in the field of vision of the camera.
- `void draw_object(Object3D * const) const`
Draws all sides of the object given as argument that are facing the camera.
- `void draw_wire_triangle(const Triangle &) const`
Draws the face given as argument (the three edges of the triangle).
- `void draw_edge(const Point<float, 4> &, const Point<float, 4> &) const`
Draws the segment given as argument.
- `Point<float, 4> perspective_projection(const Point<float, 4> &) const`
Projects the point given as argument on the screen (“near plane”).

Class Camera: Implements the camera. The camera contains:

- The height and width of the image (screen).
- The displacement speed.
- The orientation changing speed.
- The zoom speed.
- Its current position.
- Its current orientation.
- Its current displacement speed (a value for each axis x , y and z).
- Its current orientation changing speed (also three values).
- Its current zoom speed.
- The current distance between the camera and the projection plan.
- Its field of vision (frustum).

This class must also define at least the following methods:

- `move...()`
- `turn...()`

- `zoom_...()`
- `stop_move...()`
- `stop_turn...()`
- `stop_zoom...()`
- `Transformation get_transform() const`
Returns the transform corresponding to the viewpoint of the camera.
- `bool outside_frustum(const Sphere &) const`
Returns if the sphere given as argument is outside of the field of view of the camera.
- `bool sees(Triangle &) const`
Returns if the camera “sees” the triangular face given as argument.
- `LineSegment visible_part(const LineSegment &) const`
Returns the visible part of the segment given as argument.
- `void update()`
Updates the position and orientation of the camera.

Class Object3D: Implements a 3D object. It contains:

- A name.
- A position.
- A list of vertices.
- A list of faces (the mesh).

This class must, at least, define the following methods.

- `Sphere bsphere() const`
Returns the bounding sphere.
- `Triangle face(unsigned int) const`
Returns the n -th face of the object, where n is given as argument.
- `unsigned int num_faces() const`
Returns the number of faces of the object.
- `void add_face(unsigned int, unsigned int, unsigned int)` Adds a face to the object. The three integers given as arguments correspond to three vertices.
- `void remove_face(unsigned int)` Deletes a face from the object. The integer given as argument refers to the list of faces.

Class Frustum: Implements the field of view of the camera. It contains 6 planes:

- Near plane.
- Far plane.
- Right plane.
- Left plane.
- Top plane.
- Bottom plane.

This class must, at least, define the following methods:

- `bool outside(const Point<float, 4> &) const`
Returns if the point given as argument is outside the field of vision.
- `bool outside(const Sphere &) const`
Returns if the sphere given as argument is completely outside the field of vision.
- `LineSegment inter(const LineSegment &) const`
Returns the intersection between the segment and the field of vision (visible part).
- `void update(float h, float v, float e)`
Updates the position of the field of vision, where h is the horizontal resolution, v is the vertical resolution and e is the distance between the projection plane and the camera.

Main Module: In the main module of the program, you must implement two functions:

- `void load_geo_file(const * char file, Scene & scene)`
Opens a file in `.geo` format and inserts the object in the scene.
- `int main(int argc, const char * argv[])`
Initialises the GUI, reads the file (or files) in `.geo` format given as argument, executes the `main_loop` and closes the GUI. The function must also capture eventual exceptions and treat them, if possible.